SOC3210 系列

SOC3210W

SOC3210M

SOC3210I

User Manual

(含寄存器及芯片功能的详细描述)

编　者	
日　期	2008 年 8 月
版本号	V1.2

版 本 记 录

版本号	编者	日期	概要
1.0		2007 年 6 月 4 日	SOC3210 系列数据手册初稿 （根据 3210A 数据手册修改）
1.1		2007 年 7 月 10 日	更新有关 MAC、INT、GPIO、AC97、系统时钟等寄存器描述；增加 SOC3210W、SOC3210M、SOC3210B 的分类。
1.11		2007 年 7 月 23 日	修正 MAC 寄存器文字描述错误；增加 SMI 协议时序；增加 Nand flash 的地址使用描述
1.2		2007 年 8 月 5 日	将 SOC3210B 改为 SOC3210I；更改 SOC3210W 的 Nand flash

目 录

图 目 录

表 目 录

1 概述

SOC3210 系列芯片是高集成度的高性能系统级芯片（System-On-Chip），能够满足家庭网关（SOHO/SME Gateway）、NAT 路由(NAT Router)、WLAN 无线接入(WLAN access point)、VPN 网关(VPN Gateway)、VOIP 等应用的功能需求，可以应用于网络音响、数码相框、音乐播放器、VOIP 设备方案，也可应用于家庭网关、高速工业控制器等设计。

该系列芯片内嵌高性能 32 位嵌入式 RISC CPU 核，支持通用 MIPS32 指令集，主频可达 266MHz。网络方面内置 MAC，提供 MII 接口；存储器接口方面，芯片同时支持 SDRAM 接口、NOR FLASH/ROM、NAND FLASH 接口，并特置 HPI 接口可直接与 VOIP CODEC 芯片相连；提供丰富的其他外设接口支持，包括 I2C 接口、UART 串口、SPI 接口、AC97 等接口设备，提供丰富的 GPIO 接口，能够为“网络+语音”以及工业控制应用提供高效的单芯片解决方案。

SOC3210 系列包括三款芯片，分别为 SOC3210W、SOC3210M、SOC3210I，芯片外围功能和接口配置则针对不同的市场需求和应用各有侧重，总体上 SOC3210W 和 SOC3210M 功能相对互补，而 SOC3210I 则包括 SOC3210W 和 SOC3210M 的全部功能。其典型应用分别如下：

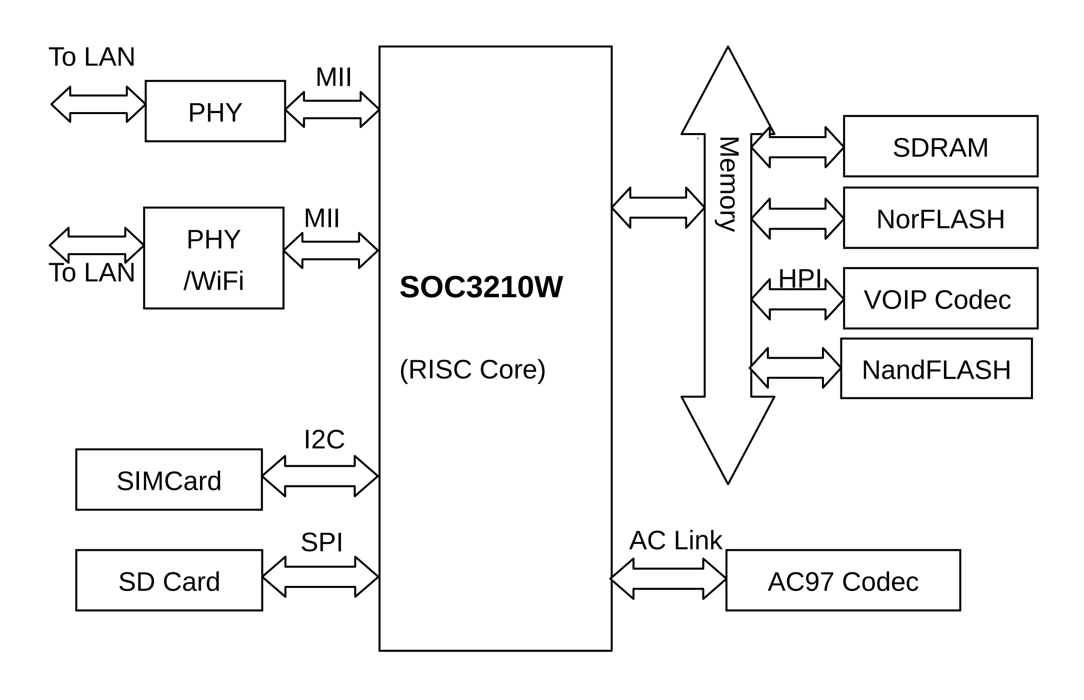


图 1-1 SOC3210W 典型应用

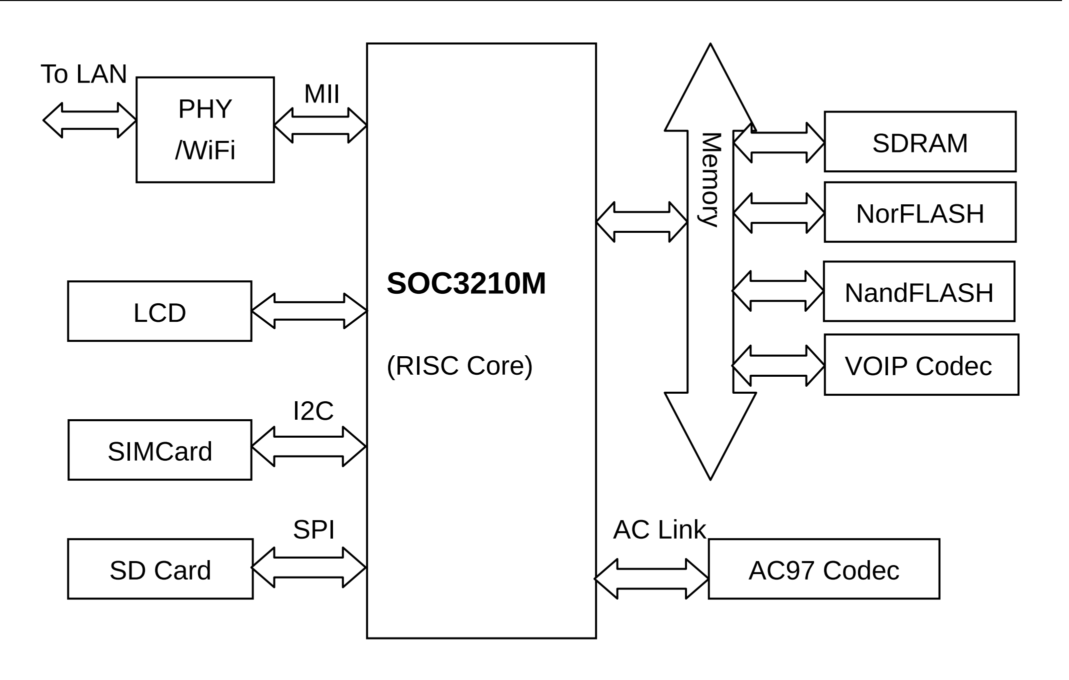


图 1-2 SOC3210M 典型应用

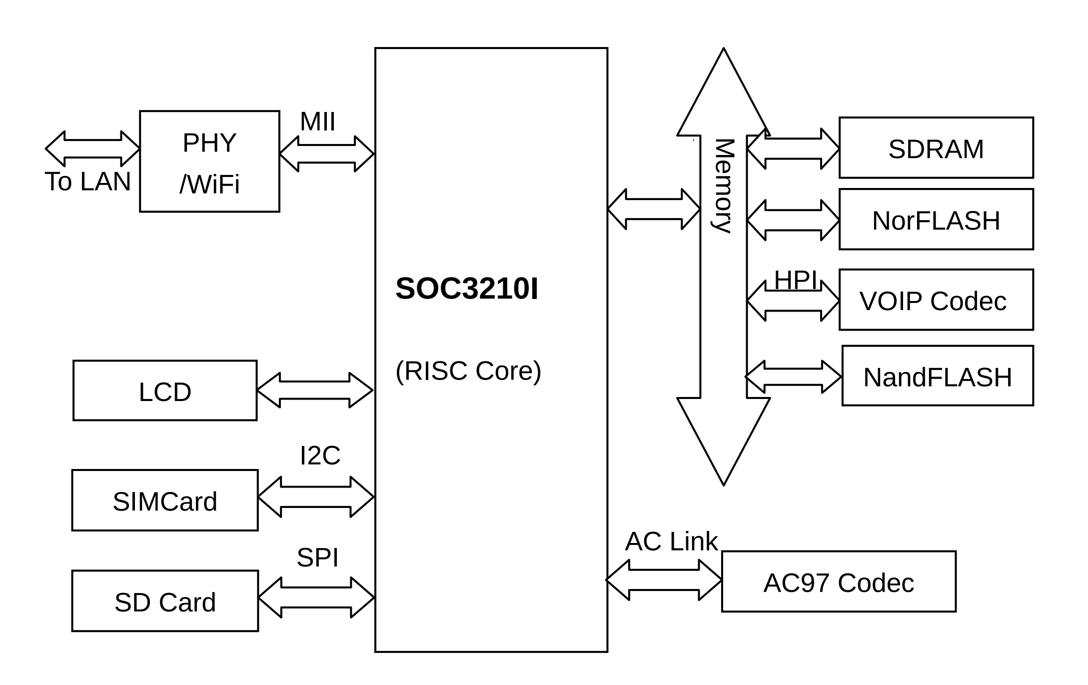


图 1-3 SOC3210I 典型应用

这三款芯片的功能概况和区别如下表：

表 1-1 SOC3210 系列芯片的功能列表

Name	RISC CPU	EMI			LCD	AC97	MII	UART	SPI	I2C	PS2	JTAG	GPIO	INT	Others	Package
		SDR	Flash	HPI												
SOC3210 W	√ 266M	√	NOR NAND	√	–	√	2	1	√	√	–	√	12	4	WDT PLL	QFP208 LQFP208
SOC3210 M	√ 266M	√	NOR NAND	√	√	√	1	1	√	√	–	√	10	4	WDT PLL	QFP208 LQFP208
SOC3210 I	√ 266M	√	NOR NAND	√	√	√	1	2	√	√	2	√	12	4	WDT PLL	QFP208 LQFP208

注： "√" 代表有一个该模块或者接口；"-" 代表没有该模块或者接口；"1"或者"2"代表有该模块或者接口的数量。

1.1 体系结构框图

本系列芯片的体系结构框图如下图所示：

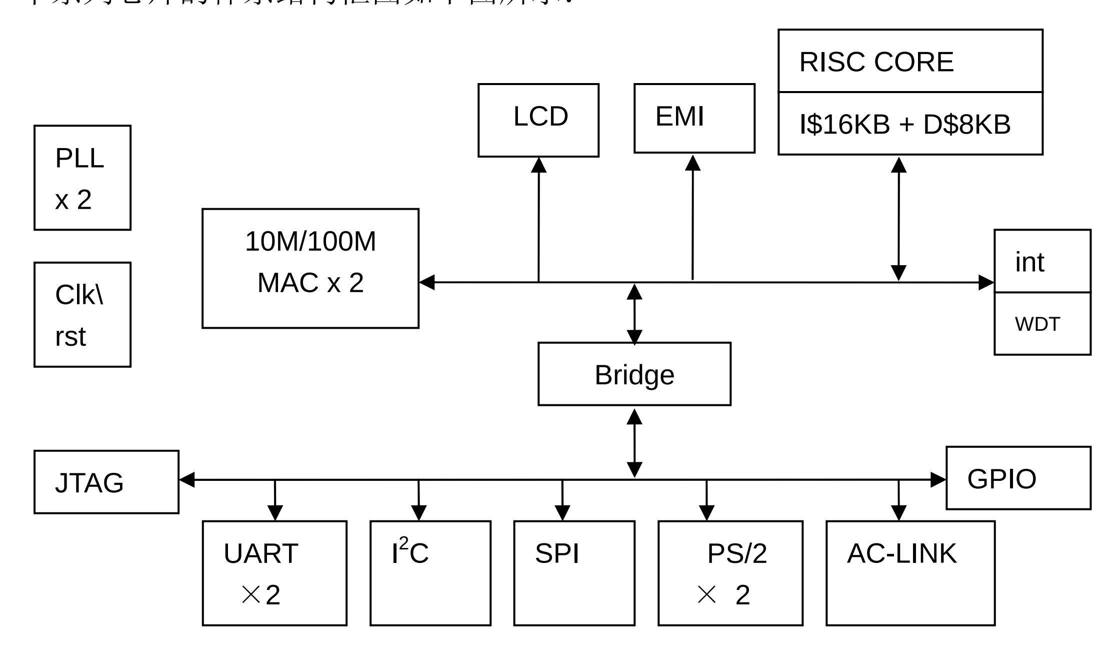


图 1-4 SOC3210 系列芯片体系结构框图

具体到 SOC3210W、SOC3210M、SOC3210I 的内部功能模块的差异请“参考表 1-1 SOC3210 系列芯片的功能列表”。

1.2 芯片主要功能

SOC3210 系列支持以下功能。

1.2.1 精简的 32 位 RISC CPU 核

✓ 32 位 RISC CPU，支持通用 MIPS32 指令集

- ✓ 五级动态流水线
- ✓ 266MHz 峰值频率
- ✓ 8 项操作队列
- ✓ 静态转移预测
- ✓ 全流水的乘法功能部件
- ✓ 32 项 TLB
- ✓ 16KB 四路组相联指令 Cache
- ✓ 8KB 两路组相联数据 Cache
- ✓ 支持非阻塞的访存

1.2.2 片上总线支持

- ✓ 分为高速系统总线（High-speed System Bus, HSB）和低速外设总线（Low-speed Peripheral Bus, LPB），中间由桥接电路相连
- ✓ 高速总线 32 位地址，32 位数据宽度
- ✓ 高速总线支持突发（burst）式传输
- ✓ 高速总线仲裁支持最高优先级
- ✓ 高速总线支持单个传输的最大传输数据量以及最多传输时间的控制
- ✓ 低速总线 10 位地址，8 位数据宽度
- ✓ 低速总线不支持突发式传输
- ✓ 低速总线时钟与系统时钟同相位，为系统时钟的 1、2、4、8、16 分频，甚至时钟停止

1.2.3 SDRAM 支持

- ✓ 最大支持 256MB
- ✓ 兼容 PC100/133 内存规范，工作频率与系统总线的时钟同频
- ✓ 可编程的突发长度：1，2，4，8 字节与整页
- ✓ 支持顺序（sequential）访问
- ✓ 字节使能访问
- ✓ 支持 SDRAM 的模式寄存器配置
- ✓ 方便灵活的访问 SDRAM 的主接口
- ✓ 透明的 SDRAM 刷新控制
- ✓ 透明的 SDRAM 换行控制
- ✓ 在串行方式读写时，连续的块数据访问之间无需额外周期的开销

1.2.4 NOR FLASH 支持

- ✓ NOR FLASH 最大支持 64MB
- ✓ 数据宽度可以通过配置设为×8 或×16bit
- ✓ 支持以块为单位的写、擦除

- ✓ 支持以字节、半字和字为单位的读数据和读产品 ID
- ✓ 对于不同厂家的 NOR FLASH，其速度可进行配置
- ✓ 低功耗设计考虑，没有命令时，Flash 及控制器自动进入待机模式
- ✓ 支持 Automatic Sleep Mode，使用此模式可以使 Flash 进一步降低功耗

1.2.5 NAND FLASH 支持

- ✓ 能支持各种容量的 NAND FLASH，地址理论极限超过 1Tera Byte
- ✓ 数据宽度可以通过配置设为×8 或×16bit
- ✓ 地址周期支持 3/4/5
- ✓ 页面大小支持 512Byte/2048Byte
- ✓ 支持块擦除、整页编程
- ✓ 支持以字节、半字和字和整页为单位的读数据、读产品 ID 和读设备状态
- ✓ 对于不同厂家的 NAND FLASH，其速度可进行配置
- ✓ 低功耗设计考虑，没有命令时，Flash 及控制器自动进入待机模式

1.2.6 HPI 接口支持

- ✓ VINETIC 芯片接口兼容
- ✓ 支持 Intel Demultiplexed Mode
- ✓ 支持 Motorola Mode

1.2.7 液晶显示控制器 LCD

SOC3210W	不支持
SOC3210M	支持
SOC3210I	支持

- ✓ 彩色/单色/灰度 LCD 面板
- ✓ 专用 DMA 视频数据通道
- ✓ 支持多种分辨率，最大 800×600 分辨率
- ✓ 支持 16/8/4/2/1 位色深
- ✓ 单色 STN 16 个灰度等级，彩色 STN 4096 色
- ✓ TFT 65536 色，支持 8 位伪彩

1.2.8 以太网 MII 接口支持

SOC3210W	支持 2 个 MII 接口
SOC3210M	支持 1 个 MII 接口
SOC3210I	支持 1 个 MII 接口

- ✓ 支持中断方式产生事件
- ✓ 每个端口均内置 MAC，完成 IEEE 802.3 MAC 层功能
  - ■ 半双工/全双工自适应
  - ■ 10/100Mbps 自适应
  - ■ 半双工时，支持碰撞检测与重发（CSMA/CD）协议

■ 支持 CRC 校验码的自动生成与校验，支持前置符生成与删除

1.2.9 AC97 接口

- ✓ 支持可变速率，最高达 48KHz
- ✓ 支持 16，18 和 20 位采样精度
- ✓ 2 频道立体声输出
- ✓ 支持麦克风输入
- ✓ 支持外部 DMA 机制

1.2.10 串行外围设备控制器 SPI

- ✓ 全双工同步串口数据传输
- ✓ 支持到 4 个的变长字节传输
- ✓ 主模式支持
- ✓ 模式故障产生错误标志并发出中断请求
- ✓ 双缓冲接收器
- ✓ 极性和相位可编程的串行时钟
- ✓ 可在等待模式下对 SPI 进行控制

1.2.11 键盘/鼠标控制器 PS/2

SOC3210W	不支持
SOC3210M	不支持
SOC3210I	支持 1 路鼠标和 1 路键盘

- ✓ 16 位可编程 5us 时钟计数器，8 位可编程 60us 时钟计数器
- ✓ 支持 PS/2 键盘接口
- ✓ 支持第一套，第二套键盘扫描码
- ✓ 支持编码键盘和非编码键盘
- ✓ 支持 PS/2 鼠标接口
- ✓ 支持二键式、三键式鼠标

1.2.12 通用异步收发器 UART

SOC3210W	支持 1 路 UART
SOC3210M	支持 1 路 UART
SOC3210I	支持 2 路 UART

- ✓ 1 路全双工异步数据接收/发送
- ✓ 可编程的数据格式
- ✓ 16 位可编程时钟计数器
- ✓ 支持接收超时检测
- ✓ 带仲裁的多中断系统
- ✓ 仅工作在 FIFO 方式
- ✓ 仅支持两个引脚（TXD，RXD）

- ✓ 在寄存器与功能上兼容 NS16550A

1.2.13 串行通讯总线控制器 I^2C

- ✓ 履行双向同步串行协议
- ✓ 只实现主设备操作
- ✓ 能够支持多主设备的总线
- ✓ 总线的时钟频率可编程
- ✓ 可以产生开始/停止/应答等操作
- ✓ 能够对总线的状态进行探测
- ✓ 与PHILIPS I^2C标准相兼容
- ✓ 支持低速和快速模式
- ✓ 支持7位寻址和10位寻址
- ✓ 支持时钟延伸和等待状态

1.2.14 通用输入/输出控制器 GPIO

SOC3210W	支持 12 路独立 GPIO
SOC3210M	支持 10 路独立 GPIO
SOC3210I	支持 12 路独立 GPIO

- ✓ 独立 GPIO 的位数多达 12 位，复用 GPIO 的位数多达 48 位，合计 60 位
- ✓ 支持位操作

1.2.15 中断控制器 Interrupt

- ✓ 支持 4 路外部中断，与 GPIO12~GPIO15 复用
- ✓ 支持软件设置中断
- ✓ 支持电平与边沿触发
- ✓ 支持中断屏蔽与使能

1.2.16 看门狗 Watchdog

- ✓ 32 比特计数器及初始化寄存器

1.2.17 其它

- ✓ 测试访问口控制器 JTAG
- ✓ 扫描测试/内建自测试 Scan/BIST
- ✓ 多时钟域系统
- ✓ 2 个锁相环 PLL，分别提供独立时钟给 CPU 和系统总线
- ✓ 芯片复位

2 接口定义

2.1 引脚与封装

SOC3210W	LQFP208 和 QFP208（两种封装仅厚度及引脚长度不同）
SOC3210M	LQFP208 和 QFP208（两种封装仅厚度及引脚长度不同）
SOC3210I	LQFP208 和 QFP208（两种封装仅厚度及引脚长度不同）

本系列芯片的封装引脚都不相同，具体封装引脚图请分别参考 SOC3210W、SOC3210M、SOC3210I 的产品 datasheet 的“引脚定义”章节。本手册只描述接口信号定义，不涉及到具体信号对应于某种封装的具体位置。

2.2 EMI 接口定义

EMI 包括 4 部分接口，分别是 HPI、NOR flash、NAND flash 和 SDRAM，除了控制信号之外，其它数据和地址信号复用。其中“复用”栏有“Y”标志的信号还特别用于上电复用的跳线引脚用，其详细功能参考后面的“系统特性”章节的描述。

对于 SDRAM 颗粒而言，有一个 CKE 引脚用于使能 SDRAM 颗粒，在与本芯片的板级连接使用时，只需将颗粒的 CKE 引脚做上拉处理即可。

SOC3210W	75PIN， 支持 SDR、Nor Flash、Nand Flash、HPI
SOC3210M	75PIN， 支持 SDR、Nor Flash、Nand Flash、HPI
SOC3210I	75PIN， 支持 SDR、Nor Flash、Nand Flash、HPI

另外有 9 个 GPIO，即 GPIO29~GPIO37 与 EMI 的地址信号 A16~A24 复用；上电后默认为地址信号，只有设置相应寄存器后才作为 GPIO 信号。

表 2-1 EMI 接口定义

No.	信号名称	类型	复用	信号描述
1	EMI_WE_	O		SDRAM/NOR-FLASH/HPI 外部存储器写使能信号
2	SDCLK	O		SDRAM 时钟
3	SD_CS_	O		SDRAM 片选信号
4	RAS_	O		SDRAM 行地址选通信号
5	CAS_	O		SDRAM 列地址选通信号
6	DQM0	O		SDRAM 数据宽度屏蔽信号
7	DQM1	O		SDRAM 数据宽度屏蔽信号
8	DQM2	O		SDRAM 数据宽度屏蔽信号
9	DQM3	O		SDRAM 数据宽度屏蔽信号
10	NR_CS_	O		NOR FLASH 片选信号
11	HPI_CS_	O		HPI 片选信号(VINETIC-2CPE/-1CPE)
12	EMI_OE_	B(O)	Y	NOR-FLASH/HPI 外部存储器输出使能信号 /CPUCLK 倍频选择 0
13	HPI_RST_	B(O)	Y	HPI 复位信号 /CPUCLK 倍频选择 2

14	HPI_RDY_	I		HPI 就绪状态信号
15	NN_CS_	O		NAND FLASH 片选信号
16	NN_RDY	I		NAND FLASH 就绪状态信号
17	NN_ALE	B	Y	NAND FLASH 地址锁存使能信号 /NORFLASH 位宽(上电时上拉表示 16 位/下拉表示 8 位）
18	NN_CLE	B	Y	NANA FLASH 命令锁存使能信号 /CPUCLK 倍频选择 1
19	A0	O		SDRAM/NORFLASH/HPI 外部存储地址总线第 0 位
20	A1	O		SDRAM/NORFLASH/HPI 外部存储地址总线第 1 位
21	A2	O		SDRAM/NORFLASH/HPI 外部存储地址总线第 2 位
22	A3	O		SDRAM/NORFLASH/HPI 外部存储地址总线第 3 位
23	A4	O		SDRAM/NORFLASH/HPI 外部存储地址总线第 4 位
24	A5	O		SDRAM/NORFLASH/HPI 外部存储地址总线第 5 位
25	A6	O		SDRAM/NORFLASH/HPI 外部存储地址总线第 6 位
26	A7	O		SDRAM/NORFLASH/HPI 外部存储地址总线第 7 位
27	A8	O		SDRAM/NORFLASH/HPI 外部存储地址总线第 8 位
28	A9	O		SDRAM/NORFLASH/HPI 外部存储地址总线第 9 位
29	A10	O		SDRAM/NORFLASH/HPI 外部存储地址总线第 10 位
30	A11	O		SDRAM/NORFLASH/HPI 外部存储地址总线第 11 位
31	A12	O		SDRAM/NORFLASH/HPI 外部存储地址总线第 12 位
32	A13	O		SDRAM/NORFLASH/HPI 外部存储地址总线第 13 位
33	A14	O		SDRAM/NORFLASH/HPI 外部存储地址总线第 14 位
34	A15	O		SDRAM/NORFLASH/HPI 外部存储地址总线第 15 位
35	A16	B (O)	GPIO29 (Y)	SDRAM/NORFLASH/HPI 外部存储地址总线第 16 位 /SDCLK 倍频选择 0
36	A17	B (O)	GPIO30 (Y)	SDRAM/NORFLASH/HPI 外部存储地址总线第 17 位 /SDCLK 倍频选择 1
37	A18	B (O)	GPIO31 (Y)	SDRAM/NORFLASH/HPI 外部存储地址总线第 18 位 /SDCLK 倍频选择 2
38	A19	B (O)	GPIO32	SDRAM/NORFLASH/HPI 外部存储地址总线第 19 位
39	A20	B (O)	GPIO33	SDRAM/NORFLASH/HPI 外部存储地址总线第 20 位
40	A21	B (O)	GPIO34	SDRAM/NORFLASH/HPI 外部存储地址总线第 21 位
41	A22	B (O)	GPIO35	SDRAM/NORFLASH/HPI 外部存储地址总线第 22 位
42	A23	B (O)	GPIO36	SDRAM/NORFLASH/HPI 外部存储地址总线第 23 位
43	A24	B (O)	GPIO37	SDRAM/NORFLASH/HPI 外部存储地址总线第 24 位
44	D0	B		SDRAM/NAND/NORFLASH/HPI 外部存储数据总线第 0 位
45	D1	B		SDRAM/NAND/NORFLASH/HPI 外部存储数据总线第 1 位
46	D2	B		SDRAM/NAND/NORFLASH/HPI 外部存储数据总线第 2 位
47	D3	B		SDRAM/NAND/NORFLASH/HPI 外部存储数据总线第 3 位
48	D4	B		SDRAM/NAND/NORFLASH/HPI 外部存储数据总线第 4 位
49	D5	B		SDRAM/NAND/NORFLASH/HPI 外部存储数据总线第 5 位
50	D6	B		SDRAM/NAND/NORFLASH/HPI 外部存储数据总线第 6 位
51	D7	B		SDRAM/NAND/NORFLASH/HPI 外部存储数据总线第 7 位

52	D8	B		SDRAM/NAND/NORFLASH/HPI 外部存储数据总线第 8 位
53	D9	B		SDRAM/NAND/NORFLASH/HPI 外部存储数据总线第 9 位
54	D10	B		SDRAM/NAND/NORFLASH/HPI 外部存储数据总线第 10 位
55	D11	B		SDRAM/NAND/NORFLASH/HPI 外部存储数据总线第 11 位
56	D12	B		SDRAM/NAND/NORFLASH/HPI 外部存储数据总线第 12 位
57	D13	B		SDRAM/NAND/NORFLASH/HPI 外部存储数据总线第 13 位
58	D14	B		SDRAM/NAND/NORFLASH/HPI 外部存储数据总线第 14 位
59	D15	B		SDRAM/NAND/NORFLASH/HPI 外部存储数据总线第 15 位
60	D16	B		SDRAM/NAND/NORFLASH/HPI 外部存储数据总线第 16 位
61	D17	B		SDRAM/NAND/NORFLASH/HPI 外部存储数据总线第 17 位
62	D18	B		SDRAM/NAND/NORFLASH/HPI 外部存储数据总线第 18 位
63	D19	B		SDRAM/NAND/NORFLASH/HPI 外部存储数据总线第 19 位
64	D20	B		SDRAM/NAND/NORFLASH/HPI 外部存储数据总线第 20 位
65	D21	B		SDRAM/NAND/NORFLASH/HPI 外部存储数据总线第 21 位
66	D22	B		SDRAM/NAND/NORFLASH/HPI 外部存储数据总线第 22 位
67	D23	B		SDRAM/NAND/NORFLASH/HPI 外部存储数据总线第 23 位
68	D24	B		SDRAM/NAND/NORFLASH/HPI 外部存储数据总线第 24 位
69	D25	B		SDRAM/NAND/NORFLASH/HPI 外部存储数据总线第 25 位
70	D26	B		SDRAM/NAND/NORFLASH/HPI 外部存储数据总线第 26 位
71	D27	B		SDRAM/NAND/NORFLASH/HPI 外部存储数据总线第 27 位
72	D28	B		SDRAM/NAND/NORFLASH/HPI 外部存储数据总线第 28 位
73	D29	B		SDRAM/NAND/NORFLASH/HPI 外部存储数据总线第 29 位
74	D30	B		SDRAM/NAND/NORFLASH/HPI 外部存储数据总线第 30 位
75	D31	B		SDRAM/NAND/NORFLASH/HPI 外部存储数据总线第 31 位

2.3 LCD 接口定义

SOC3210W	不支持 LCD
SOC3210M	支持 LCD，22 PIN
SOC3210I	支持 LCD，22 PIN

另外有 22 个 GPIO，即 GPIO39~GPIO60 与 LCD 接口复用，上电后默认为 LCD 信号，只有设置相应寄存器后才作为 GPIO 信号。

表 2-2 LCD 接口定义

No.	信号名称	类型	复用	信号描述
1	VCLK	O	GPIO39	LCD 外部时钟
2	VFRAME	B (O)	GPIO40	LCD 帧扫描信号
3	VLINE	B (O)	GPIO41	LCD 行扫描信号
4	VM	B (O)	GPIO42	LCD 可视使能信号
5	VD0	B (O)	GPIO43	LCD 像素数据第 0 位
6	VD1	B (O)	GPIO44	LCD 像素数据第 1 位
7	VD2	B (O)	GPIO45	LCD 像素数据第 2 位

8	VD3	B (O)	GPIO46	LCD 像素数据第 3 位
9	VD4	B (O)	GPIO47	LCD 像素数据第 4 位
10	VD5	B (O)	GPIO48	LCD 像素数据第 5 位
11	VD6	B (O)	GPIO49	LCD 像素数据第 6 位
12	VD7	B(O)	GPIO50	LCD 像素数据第 7 位
13	VD8	B(O)	GPIO51	LCD 像素数据第 8 位
14	VD9	B(O)	GPIO52	LCD 像素数据第 9 位
15	VD10	B (O)	GPIO53	LCD 像素数据第 10 位
16	VD11	B (O)	GPIO54	LCD 像素数据第 11 位
17	VD12	B (O)	GPIO55	LCD 像素数据第 12 位
18	VD13	B (O)	GPIO56	LCD 像素数据第 13 位
19	VD14	B (O)	GPIO57	LCD 像素数据第 14 位
20	VD15	B (O)	GPIO58	LCD 像素数据第 15 位
21	VD16	B (O)	GPIO59	LCD 像素数据第 16 位
22	VD17	B(O)	GPIO60	LCD 像素数据第 17 位

2.4 以太网 MII 接口定义

SOC3210W	17 x 2 PIN， 支持 2 路 MII 接口
SOC3210M	17 x 1 PIN， 支持 1 路 MII 接口，即 MII0；不支持 MII1 接口
SOC3210I	17 x 1 PIN， 支持 1 路 MII 接口，即 MII0；不支持 MII1 接口

表 2-3 以太网 MII0 接口定义

No.	信号名称	类型	复用	信号描述
1	M0_TXCLK	I		发送数据时钟
2	M0_TXEN	O		发送数据使能信号
3	M0_TXD0	O		发送数据第 0 位
4	M0_TXD1	O		发送数据第 1 位
5	M0_TXD2	O		发送数据第 2 位
6	M0_TXD3	O		发送数据第 3 位
7	M0_RXCLK	I		接收数据时钟
8	M0_RXDV	I		接收数据有效的状态信号
9	M0_RXERR	I		接收数据错误的状态信号
10	M0_RXD0	I		接收数据第 0 位
11	M0_RXD1	I		接收数据第 1 位
12	M0_RXD2	I		接收数据第 2 位
13	M0_RXD3	I		接收数据第 3 位
14	M0_COLL	I		发送与接收数据冲突的状态信号
15	M0_CRS	I		载波状态信号
16	M0_DIO	B		控制字串行数据
17	M0_DC	O		控制字时钟(3210B 删掉)

注：忽略 M0_TXERR 发送数据错误的状态信号

表 2-4 以太网 MII1 接口定义

No.	信号名称	类型	复用	信号描述
1	M1_TXCLK	I		发送数据时钟
2	M1_TXEN	O		发送数据使能信号
3	M1_TXD0	O		发送数据第 0 位
4	M1_TXD1	O		发送数据第 1 位
5	M1_TXD2	O		发送数据第 2 位
6	M1_TXD3	O		发送数据第 3 位
7	M1_RXCLK	I		接收数据时钟
8	M1_RXDV	I		接收数据有效的状态信号
9	M1_RXERR	I		接收数据错误的状态信号
10	M1_RXD0	I		接收数据第 0 位
11	M1_RXD1	I		接收数据第 1 位
12	M1_RXD2	I		接收数据第 2 位
13	M1_RXD3	I		接收数据第 3 位
14	M1_COLL	I		发送与接收数据冲突的状态信号
15	M1_CRS	I		载波状态信号
16	M1_DIO	B		控制字串行数据
17	M1_DC	O		控制字时钟

注：忽略 M1_TXERR 发送数据错误的状态信号

2.5 AC97 接口定义

SOC3210W、SOC3210M、SOC3210I 均支持 AC97；另有 5 个 GPIO，即 GPIO24~GPIO28 与 AC97 接口复用，上电后默认为 AC97 信号，只有设置相应寄存器后才作为 GPIO 信号。

表 2-5 AC97 接口定义

No.	信号名称	类型	复用	信号描述
1	BIT_CLK_I	B(I)	GPIO24	AC97 时钟
2	SDATA_IN	B(I)	GPIO25	AC97 数据输入
3	SDATA_OUT	B(O)	GPIO26	AC97 数据输出
4	SYNC_O	B(O)	GPIO27	AC97 输出同步信号
5	ACRESET_O	B(OD)	GPIO28	AC97 输出复位信号

2.6 SPI 接口定义

SOC3210W、SOC3210M、SOC3210I 均支持 SPI Master，另有 4 个 GPIO，即 GPIO20~GPIO23 与 SPI 接口复用，上电后默认为 SPI 信号，只有设置相应寄存器后才作为 GPIO 信号。

表 2-6 SPI 接口定义

No.	信号名称	类型	复用	信号描述
1	SCLK	B(O)	GPIO20	SPI 时钟（芯片只实现主模块）
2	MOSI	B(O)	GPIO21	SPI 主出从入数据

3	MISO	B(I)	GPIO22	SPI 主入从出数据
4	SS_	B(O)	GPIO23	SPI 选通信号

2.7 PS/2 接口定义

SOC3210W	不支持
SOC3210M	不支持
SOC3210I	支持 1 路鼠标和 1 路键盘，共 4 PIN

另有 4 个 GPIO，即 GPIO16~GPIO19 与 PS/2 接口复用，上电后默认为 PS/2 信号，只有设置相应寄存器后才作为 GPIO 信号。

表 2-7 PS/2 接口定义

No.	信号名称	类型	复用	信号描述
1	K_CLK	B (BoD)	GPIO16	键盘时钟
2	K_DAT	B (BoD)	GPIO17	键盘数据
3	M_CLK	B (BoD)	GPIO18	鼠标时钟
4	M_DAT	B (BoD)	GPIO19	鼠标数据

2.8 UART 接口定义

SOC3210W	支持 1 路 UART，共 2 PIN，即 URT0；不支持 URT1
SOC3210M	支持 1 路 UART，共 2 PIN，即 URT0；不支持 URT1
SOC3210I	支持 2 路 UART，共 4 PIN

表 2-8 UART 接口定义

No.	信号名称	类型	复用	信号描述
1	URT0_TXD	O		UART0 发送数据
2	URT0_RXD	I		UART0 接收数据
3	URT1_TXD	O		UART1 发送数据
4	URT1_RXD	I		UART1 接收数据

2.9 I^2C 接口定义

SOC3210W、SOC3210M、SOC3210I 均支持 I^2C。

表 2-9 I2C 接口定义

No.	信号名称	类型	复用	信号描述
1	I2C_SCL	BoD		I2C 时钟
2	I2C_SDA	BoD		I2C 数据

2.10 GPIO 独立接口定义

SOC3210W	支持 12 路独立 GPIO，共 12 PIN
SOC3210M	支持 10 路独立 GPIO，共 10 PIN；不支持 GPIO10~11

SOC3210I	支持 12 路独立 GPIO，共 12 PIN

表 2-10 GPIO 独立接口定义

No.	信号名称	类型	复用	信号描述
1	GPIO0	B		GPIO 专用线第 0 位
2	GPIO1	B		GPIO 专用线第 1 位
3	GPIO2	B		GPIO 专用线第 2 位
4	GPIO3	B		GPIO 专用线第 3 位
5	GPIO4	B		GPIO 专用线第 4 位
6	GPIO5	B		GPIO 专用线第 5 位
7	GPIO6	B		GPIO 专用线第 6 位
8	GPIO7	B		GPIO 专用线第 7 位
9	GPIO8	B		GPIO 专用线第 8 位
10	GPIO9	B		GPIO 专用线第 9 位
11	GPIO10	B		GPIO 专用线第 10 位，SOC3210M 不支持
12	GPIO11	B		GPIO 专用线第 11 位，SOC3210M 不支持

2.11 外部中断信号

SOC3210W、SOC3210M、SOC3210I 均支持 4 个外部中断信号。该 4 个外部中断信号也可用作普通的 GPIO12~GPIO14 信号。上电后默认屏蔽全部外部中断，只作为普通的 GPIO 信号。

表 2-11 外部中断信号定义

No.	信号名称	类型	复用	信号描述
1	INT3	B	GPIO12	GPIO 专用线第 12 位/中断 3
2	INT2	B	GPIO13	GPIO 专用线第 13 位/中断 2
3	INT1	B	GPIO14	GPIO 专用线第 14 位/中断 1
4	INT0	B	GPIO15	GPIO 专用线第 15 位/中断 0(HPI 中断）

注：4 个中断信号均内置上拉电阻。

2.12 JTAG 接口定义

SOC3210W、SOC3210M、SOC3210I 均支持 JTAG 接口，方便在线调试。

表 2-12 JTAG 接口定义

No.	信号名称	类型	复用	信号描述
1	TAP_SEL	I		TAP（测试访问口）选择（＝1 选择 EJTAG；＝0 选择 SOC），内置上拉电阻
2	JTCK	I		TAP 时钟，内置上拉电阻
3	JRST_	I		TAP 复位，内置上拉电阻
4	JTDI	I		TAP 数据输入，内置上拉电阻
5	JTDO	O		TAP 数据输出
6	JTMS	I		TAP 工作模式，内置上拉电阻

2.13 时钟/复位信号定义

表 2-13 时钟/复位信号定义

No.	信号名称	类型	复用	信号描述
1	XI	I		晶振输入(5MHz crystal)
2	XO	O		晶振输出
3	SYS_RST_	I		系统复位

2.14 电源/地信号

SOC3210 系列芯片有三组不同的电源/地，包括 PLL 模拟的电源/地、IO 电源/地和 CORE 的电源和地。SOC3210W、SOC3210M、SOC3210I 三款芯片的电源/地数量不尽相同，具体分别参考这三款产品的 datasheet。

表 2-14 PLL 电源/地信号定义

No.	电源名称	数量	描述
1	AVDD_5AP		PLL 模拟电源，1.8V (单独滤波)
2	AVSS_5AP		PLL 模拟地
3	AVDD18		PLL 模拟电源， 1.8V （单独滤波）
4	AVSS18		PLL 模拟地

表 2-15 IO 电源/地引脚定义

No.	电源名称	数量	描述
1	VCCIO		数字 IO 电源，3.3V
2	VSSIO		数字 IO 地

表 2-16 CORE 电源/地引脚定义

No.	电源名称	数量	描述
1	VCCINST		数字 core 电源，1.8V
2	VSSINST		数字 core 地

3 寄存器定义

本章给出 SOC3210 系列芯片内的控制与状态寄存器 CSR（Control & Status Register）的定义。本文约定，对于寄存器的保留位域(Reserved)，读出的值为 0，写入的值无效（即无法读出）。在引用寄存器 XXX 的位域 AAA 时，记为 XXX.AAA。

3.1 IP 内核寄存器基地址

SOC3210 系列芯片内的各个功能的控制器的寄存器的基地址如表 3-1 所示，

表 3-1 寄存器基地址

IP 内核	基地址	长度	章节	SOC3210W	SOC3210M	SOC3210I
EMI	0x1f00_0000	32	3.2	√	√	√
LCD	0x1f00_1000	4K	3.3	—	√	√
HSB MISC	0x1f00_3200	256	3.4	√	√	√
SPI	0x1f00_4000	4	3.5	√	√	√
PS/2	0x1f00_4040	32	3.6	—	—	√
UART #0	0x1f00_4080	16	3.7	√	√	√
UART #1	0x1f00_4090	16	3.7	—	—	√
I^2C	0x1f00_40D0	8	3.8	√	√	√
LPB MISC	0x1f00_4100	256	3.9	√	√	√
AC97	0x1f00_4200	128	3.10	√	√	√
AC97 DMA	0x1f00_4280	128	3.11	√	√	√
MAC #0	0x1f00_5200	256	3.12	√	√	√
MAC #1	0x1f00_5300	256	3.12	√	—	—

3.2 EMI 控制器寄存器

本模块寄存器物理地址基址为 0x1F000000。

3.2.1 SD_TIMING

中文名： SDRAM 计时寄存器
寄存器位宽： [31：0]
偏移量： 0x00
复位值： 0x0C35003F

位域	位域名称	位宽	访问	描述
31:28	Reserved	4		保留
27:16	Tref	12	R/W	刷新操作之间计数（主频 100MHz） 390： 3.9 us

位域	位域名称	位宽	访问	描述
				780： 7.8 us 1560：15.6 us 3120：31.2 us
15:11	Reserved	5		保留
10:9	tRCD	2	R/W	行地址有效到列地址有效之间计数
8:6	tRAS	3	R/W	Active 与 Precharge 命令之间计数
5:4	tRP	2	R/W	Precharge 命令执行时间计数
3:0	tRC	4	R/W	Autorefresh 与 ActiveE 命令之间计数

3.2.2 SD_MODE_SIZE

中文名：　　SDRAM 模式与大小寄存器
寄存器位宽：[31：0]
偏移量：　　0x04
复位值：　　0x00000000

位域	位域名称	位宽	访问	描述
31:30	Reserved	2		保留
29	PAA	1	R/W	初始化命令（Precharge，Autorefresh，Autorefresh）
28:24	Reserved	5		保留
23	SET_MODE	1	R/W	设置模式命令（缺省值为 0）
22	IS_SEQ	1	R/W	定义突发式读写时的块内顺序 0：顺序　　1：交替
21	Reserved	1		保留
20:19	tCAS	2	R/W	从读命令到第一个数据到来经过的计数
18:16	BLK_SIZE	3	R/W	支持多种突发块读写 000b：1 001b：2 010b：4 011b：8 111b：整页
15:8	Reserved	8		保留
7	BK_SIZE	1	R/W	每个器件中 Bank（体）的数目 0：2　　1：4
6:5	ROW_SIZE	2	R/W	行地址位数及分布 00b：A0-A10 01b：A0-A11 10b：A0-A12
4:2	COL_SIZE	3	R/W	列地址位数及分布 000b：A0-A7 001b：A0-A8 010b：A0-A9 011b：A0-A9, A11

位域	位域名称	位宽	访问	描述
				100b：A0-A9, A11, A12
1:0	BIT_SIZE	2	R/W	单个 Bank 的数据宽度 00b：4 01b：8 10b：16 11b：32

3.2.3 NR_MODE

中文名： NOR FLASH 模式配置寄存器
寄存器位宽： [31：0]
偏移量： 0x08
复位值： 0x00006A0A

位域	位域名称	位宽	访问	描述
31:16	Reserved	16		保留
15	HPI_MODE	1	R/W	HPI 由 HPI_RDY_信号响应或是由控制器计数响应 0：HPI_RDY_ 1：HPI_SPEED
14	HPI_RST_	1	R/W	HPI 接口复位输出信号
13	HPI_CS_DLY_	1	R/W	HPI_CS_工作模式: 1: MOTOROLA 2: INTEL DEMUX
12:8	HPI_SPEED	5	R/W	HPI 读写时间计数
7	NR_WIDTH	1	R/W	数据宽度 0：8 1：16
6:5	Reserved	2		保留
4:0	NR_SPEED	5	R/W	读写速度参数

3.2.4 NN_MODE

中文名： NAND FLASH 模式配置寄存器
寄存器位宽： [31：0]
偏移量： 0x0C
复位值： 0x00001315

位域	位域名称	位宽	访问	描述
31:13	Reserved	19		保留
12	NN_RDY	1	R	引脚 NN_RDY 此时的电平状态
11	Reserved	1		保留
10:8	NN_ADD_CYC	3	R/W	地址周期数 011b：3 100b：4 101b：5
7	NN_WIDTH	1	R/W	数据宽度 0：8 1：16

6:4	NN_HOLD	3	R/W	命令、地址、片选保持参数
3:0	NN_SPEED	4	R/W	读写速度参数

3.2.5 NN_CMD

中文名：　　　NAND FLASH 命令寄存器
寄存器位宽：　[31：0]
偏移量：　　　0x10
复位值：　　　0x000000FF

位域	位域名称	位宽	访问	描述
31:16	Reserved	16		保留
15:0	NN_CMD	16	-/W	命令

3.2.6 NN_ADDR

中文名：　　　NAND FLASH 地址寄存器
寄存器位宽：　[31：0]
偏移量：　　　0x14
复位值：　　　0x00000000

位域	位域名称	位宽	访问	描述
31:16	Reserved	16		保留
15:0	NN_ADDR	16	-/W	地址

3.2.7 NN_DATA

中文名：　　　NAND FLASH 数据寄存器
寄存器位宽：　[31：0]
偏移量：　　　0x40 – 0x7F
复位值：　　　0x00000000

位域	位域名称	位宽	访问	描述
31:0	NN_DATA	32	R/W	NAND 数据

3.3 LCD 控制器寄存器

本模块寄存器物理地址基址为 0x1F001000。

3.3.1 LCD_CTRL

中文名：　　　LCD 控制寄存器
寄存器位宽：　[31：0]
偏移量：　　　0x00
复位值：　　　0x00000000

位域	位域名称	位宽	访问	描述
31:26	MVAL	6	R/W	STN 模式下 VM 信号反转的频率控制 MVAL[5]=1：每帧反转一次 MVAL[5]=0：每 MVAL[4:0] x2 行反转一

位域	位域名称	位宽	访问	描述
				次
25:18	Reserved	8	R/-	保留
17	STN	1	R/W	驱动模式选择 0：VGA 模式　　1：STN 模式
16	COLOR	1	R/W	在 STN 模式下单色和彩色的选择 0：单色 STN 屏（4 位扫描宽度） 1：彩色 STN 屏（8 位扫描宽度）
15	BL	1	R/W	空闲极电平 0：正极　　1：负极
14	CSL	1	R/W	复合同步脉冲极电平 0：正极　　1：负极
13	VSL	1	R/W	垂直同步脉冲极电平 0：正极　　1：负极
12	HSL	1	R/W	水平同步脉冲极电平 0：正极　　1：负极
11	PC	1	R/W	8 位伪彩色 0：显示数据直接输出 1：经查找表输出伪彩色，此时 CD 必须为 11b
10:9	CD	2	R/W	色彩深度 11b：16 位每像素，若 PC 为 1 则 8 位 10b：4 位每像素 01b：2 位每像素 00b：1 位每像素
8:7	VBL	2	R/W	视频突发传输长度 11b：64 拍 10b：8 拍 01b：4 拍 00b：1 拍
6	CBSWE	1	R/W	颜色查找表切换控制 写 1 激活，硬件完成切换后清 0
5	VBSWE	1	R/W	帧缓冲切换控制 写 1 激活，硬件完成切换后清 0
4	CBSIE	1	R/W	颜色查找表转换中断使能，1 有效
3	VBSIE	1	R/W	帧缓冲转换中断使能，1 有效
2	HIE	1	R/W	水平同步脉冲中断使能，1 有效
1	VIE	1	R/W	垂直同步脉冲中断使能，1 有效
0	VEN	1	R/W	视频系统使能，修改其它寄存器前必须先关闭视频系统。 0：关闭 1：开启

3.3.2 LCD_STAT

中文名：　LCD 状态寄存器
寄存器位宽：　[31：0]
偏移量：　0x04
复位值：　0x00000000

位域	位域名称	位宽	访问	描述
31:20	Reserved	6	R/-	保留
17	ACMP	1	R/-	当前颜色查找表存储页号
16	AVMP	1	R/-	当前视频帧缓冲页号
15:8	Reserved	8	R/-	保留
7	CBSINT	1	R/-	颜色查找表转换中断未决（写 1 清）
6	VBSINT	1	R/-	视频存储区转换中断未决（写 1 清）
5	HINT	1	R/-	水平扫描中断未决（写 1 清）
4	VINT	1	R/-	垂直扫描中断未决（写 1 清）
3:2	Reserved	2	R/-	保留
1	LUINT	1	R/-	输出队列下溢中断未决（写 1 清）
0	SINT	1	R/-	系统错误中断未决（写 1 清）

3.3.3 LCD_HTIM

中文名：　LCD 水平计时寄存器
寄存器位宽：　[31：0]
偏移量：　0x08
复位值：　0x00000000

位域	位域名称	位宽	访问	描述
31:24	Thsync	8	R/W	水平同步脉冲宽度[1]
23:16	Thgdel	8	R/W	水平门延时
15:0	Thgate	16	R/W	水平门时域（必须能被突发长度与单次传输像素数的积整除）

3.3.4 LCD_VTIM

中文名：　LCD 垂直计时寄存器
寄存器位宽：　[31：0]
偏移量：　0x0c
复位值：　0x00000000

位域	位域名称	位宽	访问	描述
31:24	Tvsync	8	R/W	垂直同步脉冲宽度
23:16	Tvgdel	8	R/W	垂直门延时
15:0	Tvgate	16	R/W	垂直门时域

[1] 所有计数值均从 0 开始，为所需要的值减 1.

3.3.5 LCD_HVLEN

中文名：　　LCD 水平与垂直长度寄存器
寄存器位宽：[31：0]
偏移量：　　0x10
复位值：　　0x00000000

位域	位域名称	位宽	访问	描述
31:16	Thlen	16	R/W	水平长度（stn 模式下表示水平前廊长度）最小值为 Thsync+Thgdel+Thgate+3
15: 0	Tvlen	16	R/W	垂直长度

3.3.6 LCD_VBARa

中文名：　　LCD 视频基地址寄存器 a
寄存器位宽：[31：0]
偏移量：　　0x14
复位值：　　0x00000000

位域	位域名称	位宽	访问	描述
31:2	VBARa	30	R/W	视频帧缓冲 a 基地址
1:0	Reserved	2	R/-	保留

3.3.7 LCD_VBARb

中文名：　　LCD 视频基地址寄存器 b
寄存器位宽：[31：0]
偏移量：　　0x18
复位值：　　0x00000000

位域	位域名称	位宽	访问	描述
31:2	VBARb	30	R/W	视频帧缓冲 b 基地址
1:0	Reserved	2	R/-	保留

3.3.8 LCD_PCLT

中文名：　　LCD 颜色查找表寄存器
寄存器位宽：[31：0]
偏移量：　　0x800-0xffc
复位值：　　0x00000000

位域	位域名称	位宽	访问	描述
31:18	ReadREV	14	R/W	读：恒为 0；写：忽略
17:0	ColorDATA	18	R/W	颜色数据

3.4 HSB MISC 寄存器

本模块寄存器物理地址基址为 0x1F003200。

3.4.1 HSB_MISC_REG

中文名：　　　时钟分频寄存器
寄存器位宽：　[31：0]
偏移量：　　　0x00
复位值：　　　0x07EAFF08

位域	位域名称	位宽	访问	描述
31:24	Reserved	8		保留
23	EMI_HI_EN_2	1	R/W	使能地址 A[24:22]
22	EMI_HI_EN_1	1	R/W	使能地址 A[21:19]
21	EMI_HI_EN_0	1	R/W	使能地址 A[18:16]
20:18	LPB_CLK_CFG	3	R/W	LPB 总线分频设置 000: 禁止 LPB 总线时钟 001: x1/2 EXT 时钟 010: x1/4 EXT 时钟 011: x1/8 EXT 时钟 100: x1/16 EXT 时钟 111: x1 EXT 时钟
17	LCD_EN	1	R/W	在使用 LCD 时，置 1
16	Reserved	1		
15:9	LCD_CLK_CFG	7	R/W	LCD 分频设置，对 CPU_CLK 进行 LCD_CLK_CFG+1 分频
8	MUX_SEM	1	R/W	复位时为 1 对于读，取走原值，并清零 对于写，仅可将 0 写成 1
7	Reserved	1		保留
6	BOOT_SZ_CFG	1	R/-	NOR FLASH 宽度设置(上电上下拉配置)
5:3	CPU_FR_CFG	3	R/-	CPU 倍频设置(上电上下拉配置)
2:0	SD_FR_CFG	3	R/-	SDRAM 倍频设置(上电上下拉配置)

3.4.2 INT_EDGE

中文名：　　　中断触发类型寄存器
寄存器位宽：　[31：0]
偏移量：　　　0x04
复位值：　　　0x00000000

位域	位域名称	位宽	访问	描述
31:24	Reserved	8		保留
23	INT_EDGE_GPIO12	1	R/W	对应 GPIO[12]的中断,1-边延触发
22	INT_EDGE_GPIO13	1	R/W	对应 GPIO[13]的中断,1-边延触发
21	INT_EDGE_GPIO14	1	R/W	对应 GPIO[14]的中断,1-边延触发
20	INT_EDGE_GPIO15	1	R/W	对应 GPIO[15]的中断,1-边延触发
19:16	Reserved	2		保留
15	INT_EDGE_CAN1	1	R/W	对应 CAN1 的中断,1-边延触发
14	INT_EDGE_CAN0	1	R/W	对应 CAN0 的中断,1-边延触发

位域	位域名称	位宽	访问	描述
13	INT_EDGE_I2C	1	R/W	对应 I^2C 的中断,1-边延触发
12	INT_EDGE_URT1	1	R/W	对应 URT1 的中断,1-边延触发
11	INT_EDGE_URT0	1	R/W	对应 URT0 的中断,1-边延触发
10	INT_EDGE_M	1	R/W	对应 Mouse 的中断,1-边延触发
9	INT_EDGE_K	1	R/W	对应 Keyboard 的中断,1-边延触发
8	INT_EDGE_SPI	1	R/W	对应 SPI 的中断,1-边延触发
7:5	Reserved	4		保留
3	INT_EDGE_AC97	1	R/W	对应 AC97 的中断,1-边延触发
2	INT_EDGE_MAC2	1	R/W	对应 MAC2 的中断,1-边延触发
1	INT_EDGE_MAC	1	R/W	对应 MAC 的中断,1-边延触发
0	INT_EDGE_LCD	1	R/W	对应 LCD 的中断,1-边延触发

3.4.3 INT_STEER

中文名：　　　中断输出控制寄存器
寄存器位宽：　[31：0]
偏移量：　　　0x08
复位值：　　　0x00000000

位域	位域名称	位宽	访问	描述
31:24	Reserved	8		保留
23	INT_STEER_GPIO12	1	R/W	对应 GPIO[12]的中断, 0- 通过 CPU 中断 0, 1- 通过 CPU 中断 1
22	INT_STEER_GPIO13	1	R/W	对应 GPIO[13]的中断, 0- 通过 CPU 中断 0, 1- 通过 CPU 中断 1
21	INT_STEER_GPIO14	1	R/W	对应 GPIO[14]的中断, 0- 通过 CPU 中断 0, 1- 通过 CPU 中断 1
20	INT_STEER_GPIO15	1	R/W	对应 GPIO[15]的中断, 0- 通过 CPU 中断 0, 1- 通过 CPU 中断 1
19:16		4	R/W	保留
15	INT_STEER_CAN1	1	R/W	对应 CAN1 的中断, 0- 通过 CPU 中断 0, 1- 通过 CPU 中断 1
14	INT_STEER_CAN0	1	R/W	对应 CAN0 的中断, 0- 通过 CPU 中断 0, 1- 通过 CPU 中断 1
13	INT_STEER_I2C	1	R/W	对应 I^2C 的中断, 0- 通过 CPU 中断 0, 1- 通过 CPU 中断 1
12	INT_STEER_URT1	1	R/W	对应 URT1 的中断, 0- 通过 CPU 中断 0, 1- 通过 CPU 中断 1
11	INT_STEER_URT0	1	R/W	对应 URT0 的中断, 0- 通过 CPU 中断 0, 1- 通过 CPU 中断 1
10	INT_STEER_M	1	R/W	对应 Mouse 的中断, 0- 通过 CPU 中断 0, 1- 通过 CPU 中断 1
9	INT_STEER_K	1	R/W	对应 Keyboard 的中断, 0- 通过 CPU 中断 0, 1- 通过 CPU 中断 1

8	INT_STEER_SPI	1	R/W	对应 SPI 的中断, 0- 通过 CPU 中断 0, 1- 通过 CPU 中断 1
7:4	Reserved	4		保留
3	INT_STEER_AC97	1	R/W	对应 AC97 的中断, 0- 通过 CPU 中断 0, 1- 通过 CPU 中断 1
2	INT_STEER_MAC2	1	R/W	对应 MAC2 的中断, 0- 通过 CPU 中断 0, 1- 通过 CPU 中断 1
1	INT_STEER_MAC	1	R/W	对应 MAC 的中断, 0- 通过 CPU 中断 0, 1- 通过 CPU 中断 1
0	INT_STEER_LCD	1	R/W	对应 LCD 的中断, 0- 通过 CPU 中断 0, 1- 通过 CPU 中断 1

3.4.4 INT_POL

中文名： 中断极性控制寄存器
寄存器位宽： [31：0]
偏移量： 0x0C
复位值： 0x0000FF1F

位域	位域名称	位宽	访问	描述
31:24	Reserved	11		保留
23	INT_POL_GPIO12	1	R/W	对应 GPIO[12]的中断, 1 –高电平有效, 0 –低电平有效
22	INT_POL_GPIO13	1	R/W	对应 GPIO[13]的中断, 1 –高电平有效, 0 –低电平有效
21	INT_POL_GPIO14	1	R/W	对应 GPIO[14]的中断, 1 –高电平有效, 0 –低电平有效
20	INT_POL_GPIO15	1	R/W	对应 GPIO[15]的中断, 1 –高电平有效, 0 –低电平有效
19:16		4	R/W	保留
15	INT_POL_CAN1	1	R/W	对应 CAN1 的中断, 1 –高电平有效, 0 –低电平有效
14	INT_POL_CAN0	1	R/W	对应 CAN0 的中断, 1 –高电平有效, 0 –低电平有效
13	INT_POL_I2C	1	R/W	对应 I^2C 的中断, 1 –高电平有效, 0 –低电平有效
12	INT_POL_URT1	1	R/W	对应 URT1 的中断, 1 –高电平有效, 0 –低电平有效
11	INT_POL_URT0	1	R/W	对应 URT0 的中断, 1 –高电平有效, 0 –低电平有效
10	INT_POL_M	1	R/W	对应 Mouse 的中断, 1 –高电平有效, 0 –低电平有效
9	INT_POL_K	1	R/W	对应 Keyboard 的中断, 1 –高电平有效, 0 –低电平有效

8	INT_POL_SPI	1	R/W	对应 SPI 的中断, 1 –高电平有效, 0 –低电平有效
7:4	Reserved	4		保留
3	INT_POL_AC97	1	R/W	对应 AC97 的中断, 1 –高电平有效, 0 –低电平有效
2	INT_POL_MAC2	1	R/W	对应 MAC2 的中断, 1 –高电平有效, 0 –低电平有效
1	INT_POL_MAC	1	R/W	对应 MAC 的中断, 1 –高电平有效, 0 –低电平有效
0	INT_POL_LCD	1	R/W	对应 LCD 的中断, 1 –高电平有效, 0 –低电平有效

3.4.5 INT_ SET

中文名：　　　中断设置寄存器
寄存器位宽：　[31：0]
偏移量：　　　0x10
复位值：　　　0x00000000

位域	位域名称	位宽	访问	描述
31:24	Reserved	8		保留
23	INT_SET_GPIO12	1	R/W	对应 GPIO[12]的中断,1 –强置中断
22	INT_SET_GPIO13	1	R/W	对应 GPIO[13]的中断,1 –强置中断
21	INT_SET_GPIO14	1	R/W	对应 GPIO[14]的中断,1 –强置中断
20	INT_SET_GPIO15	1	R/W	对应 GPIO[15]的中断,1 –强置中断
19:16		4	R/W	保留
15	INT_SET_CAN1	1	R/W	对应 CAN1 的中断,1 –强置中断
14	INT_SET_CAN0	1	R/W	对应 URT0 的中断,1 –强置中断
13	INT_SET_I2C	1	R/W	对应 I^{2}C 的中断,1 –强置中断
12	INT_SET_URT1	1	R/W	对应 URT1 的中断,1 –强置中断
11	INT_SET_URT0	1	R/W	对应 URT0 的中断,1 –强置中断
10	INT_SET_M	1	R/W	对应 Mouse 的中断,1 –强置中断
9	INT_SET_K	1	R/W	对应 Keyboard 的中断,1 –强置中断
8	INT_SET_SPI	1	R/W	对应 SPI 的中断,1 –强置中断
7:5	Reserved	3		保留
3	INT_SET_AC97	1	R/W	对应 AC97 的中断,1 –强置中断
2	INT_SET_MAC2	1	R/W	对应 MAC2 的中断,1 –强置中断
1	INT_SET_MAC	1	R/W	对应 MAC 的中断,1 –强置中断
0	INT_SET_LCD	1	R/W	对应 LCD 的中断,1 –强置中断

3.4.6 INT_CLR

中文名： 中断使能清除寄存器
寄存器位宽： [31：0]
偏移量： 0x14
复位值： 0x00000000

位域	位域名称	位宽	访问	描述
31:21	Reserved	11		保留
23	INT_CLR_GPIO12	1	R/W	对应 GPIO[12]的中断,1 –清中断
22	INT_CLR_GPIO13	1	R/W	对应 GPIO[13]的中断,1 –清中断
21	INT_CLR_GPIO14	1	R/W	对应 GPIO[14]的中断,1 –清中断
20	INT_CLR_GPIO15	1	R/W	对应 GPIO[15]的中断,1 –清中断
19:16		4	R/W	保留
15	INT_CLR_CAN1	1	R/W	对应 CAN1 的中断,1 –清中断
14	INT_CLR_CAN0	1	R/W	对应 CAN0 的中断,1 –清中断
13	INT_CLR_I2C	1	R/W	对应 I^2C 的中断,1 –清中断
12	INT_CLR_URT1	1	R/W	对应 URT1 的中断,1 –清中断
11	INT_CLR_URT0	1	R/W	对应 URT0 的中断,1 –清中断
10	INT_CLR_M	1	R/W	对应 Mouse 的中断,1 –清中断
9	INT_CLR_K	1	R/W	对应 Keyboard 的中断,1 –清中断
8	INT_CLR_SPI	1	R/W	对应 SPI 的中断,1 –清中断
7:4	Reserved	4		保留
3	INT_CLR_AC97	1	R/W	对应 AC97 的中断,1 –清中断
2	INT_CLR_MAC2	1	R/W	对应 MAC2 的中断,1 –清中断
1	INT_CLR_MAC	1	R/W	对应 MAC 的中断,1 –清中断
0	INT_CLR_LCD	1	R/W	对应 LCD 的中断,1 –清中断

3.4.7 INT_EN

中文名： 中断使能寄存器
寄存器位宽： [31：0]
偏移量： 0x18
复位值： 0x00000000

位域	位域名称	位宽	访问	描述
31:24	Reserved	8		保留
23	INT_EN_GPIO	1	R/W	对应 GPIO[12]的中断, 1 –有效
22	INT_EN_GPIO	1	R/W	对应 GPIO[13]的中断, 1 –有效
21	INT_EN_GPIO	1	R/W	对应 GPIO[14]的中断, 1 –有效
20	INT_EN_GPIO	1	R/W	对应 GPIO[15]的中断, 1 –有效

19:16		4	R/W	保留
15	INT_EN_CAN1	1	R/W	对应 CAN1 的中断, 1 –有效
14	INT_EN_CAN0	1	R/W	对应 CAN0 的中断, 1 –有效
13	INT_EN_I2C	1	R/W	对应 I^2C 的中断, 1 –有效
12	INT_EN_URT1	1	R/W	对应 URT1 的中断, 1 –有效
11	INT_EN_URT0	1	R/W	对应 URT0 的中断, 1 –有效
10	INT_EN_M	1	R/W	对应 Mouse 的中断, 1 –有效
9	INT_EN_K	1	R/W	对应 Keyboard 的中断, 1 –有效
8	INT_EN_SPI	1	R/W	对应 SPI 的中断, 1 –有效
7:4	Reserved	4		保留
3	INT_EN_AC97	1	R/W	对应 AC97 的中断, 1 –有效
2	INT_EN_MAC2	1	R/W	对应 MAC2 的中断, 1 –有效
1	INT_EN_MAC	1	R/W	对应 MAC 的中断, 1 –有效
0	INT_EN_LCD	1	R/W	对应 LCD 的中断, 1 –有效

3.4.8 INT_ISR

中文名： 中断状态寄存器
寄存器位宽： [31：0]
偏移量： 0x1C
复位值： 0x00000000

位域	位域名称	位宽	访问	描述
31:24	Reserved	8		保留
23	INT_ISR_GPIO	1	R/W	对应 GPIO[12]的中断, 1 –有效
22	INT_ISR_GPIO	1	R/W	对应 GPIO[13]的中断, 1 –有效
21	INT_ISR_GPIO	1	R/W	对应 GPIO[14]的中断, 1 –有效
20	INT_ISR_GPIO	1	R/W	对应 GPIO[15]的中断, 1 –有效
19:14		6	R/W	保留
13	INT_ISR_I2C	1	R/W	对应 I^2C 的中断, 1 –有效
12	INT_ISR_URT1	1	R/W	对应 URT1 的中断, 1 –有效
11	INT_ISR_URT0	1	R/W	对应 URT0 的中断, 1 –有效
10	INT_ISR_M	1	R/W	对应 Mouse 的中断, 1 –有效
9	INT_ISR_K	1	R/W	对应 Keyboard 的中断, 1 –有效
8	INT_ISR_SPI	1	R/W	对应 SPI 的中断, 1 –有效
7:4	Reserved	3		保留
3	INT_ISR_AC97	1	R/W	对应 AC97 的中断, 1 –有效
2	INT_ISR_MAC2	1	R/W	对应 MAC2 的中断, 1 –有效
1	INT_ISR_MAC	1	R/W	对应 MAC 的中断, 1 –有效
0	INT_ISR_LCD	1	R/W	对应 LCD 的中断, 1 –有效

3.4.9 GPIO_OE_60_29

中文名： GPIO[60:29]输出使能寄存器
寄存器位宽： [31：0]
偏移量： 0x20

复位值： 0x00000000

位域	位域名称	位宽	访问	描述
31:0	GPIO_OE_60_29	32	R/W	GPIO 输出使能控制，1 –有效，其中 GPIO_OE_38 保留

3.4.10 GPIO_I_60_29

中文名： GPIO[60:29]引脚输入寄存器
寄存器位宽： [31：0]
偏移量： 0x24
复位值： 0x000001C0

位域	位域名称	位宽	访问	描述
31:0	GPIO_I_60_29	32	R/W	GPIO 引脚输入寄存器，其中 GPIO_I_38 保留

3.4.11 GPIO_O_60_29

中文名： GPIO[60:29]引脚输出寄存器
寄存器位宽： [31：0]
偏移量： 0x28
复位值： 0x00000000

位域	位域名称	位宽	访问	描述
31:0	GPIO_O_60_29	32	R/W	GPIO 引脚输出寄存器，其中 GPIO_O_38 保留

3.4.12 HSB_ARB_CFG

中文名： HSB 仲裁配置寄存器
寄存器位宽： [31：0]
偏移量： 0x2C
复位值： 0x003F00C8

位域	位域名称	位宽	访问	描述
31:16	HSB_TRANS_COUNT	16	R/W	HSB 总线一次请求最大传输数 FFFF：禁止按传输数打断总线操作
15:0	HSB_CYC_COUNT	16	R/W	HSB 总线一次请求最大时钟数 FFFF：禁止按时钟数打断总线操作

3.4.13 WD_TIMER

中文名： 看门狗定时寄存器
寄存器位宽： [31：0]
偏移量： 0x30
复位值： 0x00000000

位域	位域名称	位宽	访问	描述

31:0	WD_TIMER	32	R/W	看门狗定时器设置

3.4.14 WD_CTRL

中文名：　　　看门狗控制寄存器
寄存器位宽：　[31：0]
偏移量：　　　0x34
复位值：　　　0x00000000

位域	位域名称	位宽	访问	描述
31:9	Reserved	23		保留
8	WD_SIG	1	R/W	喂狗信号
7:1	Reserved	7		保留
0:0	WD_EN	1	R/W	0：看门狗禁止　　1：看门狗使能

3.5 SPI 控制器寄存器

本模块寄存器物理地址基址为 0x1F004000。

3.5.1 控制寄存器（SPCR）

中文名：　　　控制寄存器
寄存器位宽：　[7：0]
偏移量：　　　0x00
复位值：　　　0x10

位域	位域名称	位宽	访问	描述
7	Spie	1	RW	中断输出使能信号 高有效
6	spe	1	RW	系统工作使能信号高有效
5	Reserved	1	RW	保留
4	mstr	1	RW	master 模式选择位，此位一直保持 1
3	cpol	1	RW	时钟极性位
2	cpha	1	RW	时钟相位位 1 则相位相反，为 0 则相同
1:0	spr	2	RW	sclk_o 分频设定，需要与 sper 的 spre 一起使用

3.5.2 状态寄存器（SPSR）

中文名：　　　状态寄存器
寄存器位宽：　[7：0]
偏移量：　　　0x01
复位值：　　　0x00

位域	位域名称	位宽	访问	描述
7	spif	1	RW	中断标志位 1 表示有中断申请，写 1 则清零
6	wcol	1	RW	写寄存器溢出标志位 为 1 表示已经溢出，写 1 则清零
5:4	Reserved	2	RW	保留
3	wffull	1	RW	写寄存器满标志 1 表示已经满

2	wfempty	1	RW	写寄存器空标志 1 表示空
1	rffull	1	RW	读寄存器满标志 1 表示已经满
0	rfempty	1	RW	读寄存器空标志 1 表示空

3.5.3 数据寄存器（TxFIFO）

中文名： 数据传输寄存器
寄存器位宽： [7：0]
偏移量： 0x02
复位值： 0x00

位域	位域名称	位宽	访问	描述
7:0	Tx FIFO	8	W	数据传输寄存器

3.5.4 外部寄存器（SPER）

中文名： 外部寄存器
寄存器位宽： [7：0]
偏移量： 0x03
复位值： 0x00

位域	位域名称	位宽	访问	描述
7:6	icnt	2	RW	在传输完多少个字节后送出中断申请信号 00 – 1 字节 01 - 2 字节 10 - 3 字节 11 - 3 字节
5:2	Reserved	4	RW	保留
1:0	spre	2	RW	与 Spr 一起设定分频的比率

分频系数：

spre spr	00 00	00 01	00 10	00 11	01 00	01 01	01 10	01 11	10 00	10 01	10 10	10 11
分频系数	2	4	16	32	8	64	128	256	512	1024	2048	4096

3.6 PS/2 控制器寄存器

本模块寄存器物理地址基址为 0x1F004040。

3.6.1 输入缓冲区寄存器（RIBUF）

中文名： 接收缓冲区
寄存器位宽： [7：0]
偏移量： 0x00
复位值： 0x00

位域	位域名称	位宽	访问	描述
7:0	RIBUF	8	R	存放接收到的字节

3.6.2 输出缓冲区寄存器（WOBUF）

中文名：　　　输出缓冲区
寄存器位宽：　[7：0]
偏移量：　　　0x00
复位值：　　　0x00

位域	位域名称	位宽	访问	描述
7:0	WOBUF	8	R	存放发送的字节

3.6.3 状态寄存器（RSR）

中文名：　　　状态寄存器
寄存器位宽：　[7：0]
偏移量：　　　0x04
复位值：　　　0x00

位域	位域名称	位宽	访问	描述
7	PERR	1	R	奇偶校验错误位
6	TOUT	1	R	超时位
5	AIBF	1	R	鼠标输入缓冲满
4	KDI	1	R	键盘锁定状态
3	A2	1	R	命令/数据指示
2	SFLG	1	R	系统标志
1	OBF	1	R	键盘输出缓冲满
0	IBF	1	R	键盘输入缓冲满

3.6.4 命令寄存器（WSC）

中文名：　　　命令寄存器
寄存器位宽：　[7：0]
偏移量：　　　0x04
复位值：　　　0x00

位域	位域名称	位宽	访问	描述
7	Reserved	1	W	保留
6	TRANS	1	W	转换位，置 1 后，则键盘扫描码转换被启动
5	EN_MS	1	W	鼠标使能位，置 1 后，鼠标接口被禁止
4	EN_KB	1	W	键盘使能位，置 1 后，键盘接口被禁止
3	Reserved	1	W	保留
2	SYS	1	W	系统标志位，与状态寄存器 SFLG 位连接
1	IRQ_MS	1	W	鼠标中断使能
0	IRO_KB	1	W	键盘中断使能

3.6.5 分频锁存器低字节寄存器（DLL）

中文名：　　　分频锁存器低

寄存器位宽： [7：0]
偏移量： 0x08
复位值： 0x00

位域	位域名称	位宽	访问	描述
7:0	DLL	8	RW	存放分频锁存器的低 8 位

3.6.6 分频锁存器高字节寄存器（DLH）

中文名： 分频锁存器高
寄存器位宽： [7：0]
偏移量： 0x09
复位值： 0x00

位域	位域名称	位宽	访问	描述
7:0	DLH	8	RW	存放分频锁存器的高 8 位

3.6.7 键盘分频锁存器（DL_KBD）

中文名： 键盘分频锁存器
寄存器位宽： [7：0]
偏移量： 0x0A
复位值： 0x0C

位域	位域名称	位宽	访问	描述
7:0	DL_KBD	8	RW	存放键盘 60usec 分频锁存器值

3.6.8 鼠标分频锁存器（DL_AUX）

中文名： 鼠标分频锁存器
寄存器位宽： [7：0]
偏移量： 0x0B
复位值： 0x0C

位域	位域名称	位宽	访问	描述
7:0	DL_AUX	8	RW	存放鼠标 60usec 分频锁存器值

3.7 UART 控制器寄存器

本模块有两个并行工作的 UART 接口，其功能寄存器完全一样，只是访问基址不一样。

UART0 寄存器物理地址基址为 0x1F004080。

UART1 寄存器物理地址基址为 0x1F004090。

3.7.1 数据寄存器（DAT）

中文名： 数据传输寄存器
寄存器位宽： [7：0]
偏移量： 0x00

复位值： 0x00

位域	位域名称	位宽	访问	描述
7:0	Tx FIFO	8	W	数据传输寄存器

3.7.2 中断使能寄存器（IER）

中文名： 中断使能寄存器
寄存器位宽： [7：0]
偏移量： 0x01
复位值： 0x00

位域	位域名称	位宽	访问	描述
7:4	Reserved	4	RW	保留
3	IME	1	RW	Modem 状态中断使能 ‘0’ – 关闭 ‘1’ – 打开
2	ILE	1	RW	接收器线路状态中断使能 ‘0’ – 关闭 ‘1’ – 打开
1	ITxE	1	RW	传输保存寄存器为空中断使能 ‘0’ – 关闭 ‘1’ – 打开
0	IRxE	1	RW	接收有效数据中断使能 ‘0’ – 关闭 ‘1’ – 打开

3.7.3 中断标识寄存器（IIR）

中文名： 中断源寄存器
寄存器位宽： [7：0]
偏移量： 0x02
复位值： 0xc1

位域	位域名称	位宽	访问	描述
7:4	Reserved	4	R	保留
3:1	II	3	R	中断源表示位，详见下表
0	INTp	1	R	中断表示位

中断控制功能表

Bit 3	Bit 2	Bit 1	优先级	中断类型	中断源	中断复位控制
0	1	1	1st	接收线路状态	奇偶、溢出或帧错误，或打断中断	读 LSR
0	1	0	2nd	接收到有效数据	FIFO 的字符个数达到 trigger 的水平	FIFO 的字符个数低于 trigger 的值
1	1	0	2nd	接收超时	在 FIFO 至少有一个字符，但在 4 个字符时间内没有任何操作，包括读和写操作	读接收 FIFO

0	0	1	3rd	传输保存寄存器为空	传输保存寄存器为空	写数据到 THR 或者多 IIR
0	0	0	4th	Modem 状态	CTS, DSR, RI or DCD.	读 MSR

3.7.4 FIFO 控制寄存器（FCR）

中文名： FIFO 控制寄存器
寄存器位宽： [7：0]
偏移量： 0x02
复位值： 0xc0

位域	位域名称	位宽	访问	描述
7:6	TL	2	W	接收 FIFO 提出中断申请的 trigger 值 ‘00’ – 1 字节 ‘01’ – 4 字节 ‘10’ – 8 字节 ‘11’ – 14 字节
5:3	Reserved	3	W	保留
2	Txset	1	W	‘1’ 清除发送 FIFO 的内容，复位其逻辑
1	Rxset	1	W	‘1’ 清除接收 FIFO 的内容，复位其逻辑
0	Reserved	1	W	保留

3.7.5 线路控制寄存器（LCR）

中文名： 线路控制寄存器
寄存器位宽： [7：0]
偏移量： 0x03
复位值： 0x03

位域	位域名称	位宽	访问	描述
7	dlab	1	RW	分频锁存器访问位 ‘1’ – 访问操作分频锁存器 ‘0’ – 访问操作正常寄存器
6	bcb	1	RW	打断控制位 ‘1’ – 此时串口的输出被置为 0(打断状态). ‘0’ – 正常操作
5	spb	1	RW	指定奇偶校验位 ‘0’ – 不用指定奇偶校验位 ‘1’ – 如果 LCR[4]位是 1 则传输和检查奇偶校验位为 0。如果 LCR[4]位是 0 则传输和检查奇偶校验位为 1。
4	eps	1	RW	奇偶校验位选择 ‘0’ – 在每个字符中有奇数个 1（包括数据和奇偶校验位） ‘1’ –在每个字符中有偶数个 1
3	pe	1	RW	奇偶校验位使能 ‘0’ – 没有奇偶校验位

				‘1’ – 在输出时生成奇偶校验位，输入则判断奇偶校验位
2	sb	1	RW	定义生成停止位的位数 ‘0’ – 1 个停止位 ‘1’ – 在 5 位字符长度时是 1.5 个停止位，其他长度是 2 个停止位
1:0	bec	2	RW	设定每个字符的位数 ‘00’ – 5 位 ‘01’ – 6 位 ‘10’ – 7 位 ‘11’ – 8 位

3.7.6 MODEM 控制寄存器（MCR）

中文名： Modem 控制寄存器
寄存器位宽： [7：0]
偏移量： 0x04
复位值： 0x00

位域	位域名称	位宽	访问	描述
7:5	Reserved	3	W	保留
4	Loop	1	W	回环模式控制位 ‘0’ – 正常操作 ‘1’ –回环模式。在在回环模式中，TXD 输出一直为 1，输出移位寄存器直接连到输入移位寄存器中。其他连接如下。 DTR ➔ DSR RTS ➔ CTS Out1 ➔ RI Out2 ➔ DCD
3	OUT2	1	W	在回环模式中连到 DCD 输入
2	OUT1	1	W	在回环模式中连到 RI 输入
1	RTSC	1	W	RTS 信号控制位
0	DTRC	1	W	DTR 信号控制位

3.7.7 线路状态寄存器（LSR）

中文名： 线路状态寄存器
寄存器位宽： [7：0]
偏移量： 0x05
复位值： 0x00

位域	位域名称	位宽	访问	描述
7	ERROR	1	R	错误表示位 ‘1’ – 至少有奇偶校验位错误，帧错误或打断中断的一个。 ‘0’ – 没有错误
6	TE	1	R	传输为空表示位 ‘1’ – 传输 FIFO 和传输移位寄存器都为

位域	位域名称	位宽	访问	描述
				空。给传输 FIFO 写数据时清零 ‘0’ – 有数据
5	TFE	1	R	传输 FIFO 位空表示位 ‘1’ – 当前传输 FIFO 为空，给传输 FIFO 写数据时清零 ‘0’ – 有数据
4	BI	1	R	打断中断表示位 ‘1’ –接收到 起始位＋数据＋奇偶位＋停止位都是 0，即有打断中断 ‘0’ – 没有打断
3	FE	1	R	帧错误表示位 ‘1’ – 接收的数据没有停止位 ‘0’ – 没有错误
2	PE	1	R	奇偶校验位错误表示位 ‘1’ – 当前接收数据有奇偶错误 ‘0’ – 没有奇偶错误
1	OE	1	R	数据溢出表示位 ‘1’ – 有数据溢出 ‘0’ – 无溢出
0	DR	1	R	接收数据有效表示位 ‘0’ – 在 FIFO 中无数据 ‘1’ – 在 FIFO 中有数据

对这个寄存器进行读操作时，LSR[4:1]和 LSR[7]被清零，LSR[6:5]在给传输 FIFO 写数据时清零，LSR[0]则对接收 FIFO 进行判断。

3.7.8 MODEM 状态寄存器 （MSR）

中文名：　Modem 状态寄存器
寄存器位宽：　[7：0]
偏移量：　0x06
复位值：　0x00

位域	位域名称	位宽	访问	描述
7	CDCD	1	R	DCD 输入值的反，或者在回环模式中连到 Out2
6	CRI	1	R	RI 输入值的反，或者在回环模式中连到 OUT1
5	CDSR	1	R	DSR 输入值的反，或者在回环模式中连到 DTR
4	CCTS	1	R	CTS 输入值的反，或者在回环模式中连到 RTS
3	DDCD	1	R	DDCD 指示位
2	TERI	1	R	RI 边沿检测。RI 状态从低到高变化
1	DDSR	1	R	DDSR 指示位

0	DCTS	1	R	DCTS 指示位

3.7.9 分频锁存器

中文名：　　　分频锁存器 1
寄存器位宽：　[7：0]
偏移量：　　　0x00
复位值：　　　0x00

位域	位域名称	位宽	访问	描述
7:0	LSB	8	RW	存放分频锁存器的低 8 位

中文名：　　　分频锁存器 2
寄存器位宽：　[7：0]
偏移量：　　　0x01
复位值：　　　0x00

位域	位域名称	位宽	访问	描述
7:0	MSB	8	RW	存放分频锁存器的高 8 位

3.8 I^2C 控制器寄存器

本模块寄存器物理地址基址为 0x1F0040D0。

3.8.1 分频锁存器低字节寄存器（PRERlo）

中文名：　　　分频锁存器低字节寄存器
寄存器位宽：　[7：0]
偏移量：　　　0x00
复位值：　　　0xff

位域	位域名称	位宽	访问	描述
7:0	PRERlo	8	RW	存放分频锁存器的低 8 位

3.8.2 分频锁存器高字节寄存器（PRERhi）

中文名：　　　分频锁存器高字节寄存器
寄存器位宽：　[7：0]
偏移量：　　　0x01
复位值：　　　0xff

位域	位域名称	位宽	访问	描述
7:0	PRERhi	8	RW	存放分频锁存器的高 8 位

假设分频锁存器的值为 prescale，从 LPB 总线 PCLK 时钟输入的频率为 clock_a，SCL 总线的输出频率为 clock_s，则应满足如下关系：

Prcescale　=　clock_a/(5*clock_s)-1

3.8.3 控制寄存器（CTR）

中文名：　　　控制寄存器

寄存器位宽：　[7：0]
偏移量：　　　0x02
复位值：　　　0x00

位域	位域名称	位宽	访问	描述
7	EN	1	RW	模块工作使能位 为 1 正常工作模式, 0 对分频寄存器进行操作
6	IEN	1	RW	中断使能位 为 1 则打开中断
5:0	Reserved	6	RW	保留

3.8.4 发送数据寄存器（TXR）

中文名：　　　发送寄存器
寄存器位宽：　[7：0]
偏移量：　　　0x03
复位值：　　　0x00

位域	位域名称	位宽	访问	描述
7:1	DATA	7	W	存放下个将要发送的字节
0	DRW	1	W	当数据传送时，该位保存的是数据的最低位； 当地址传送时，该位指示读写状态

3.8.5 接受数据寄存器（RXR）

中文名：　　　接收寄存器
寄存器位宽：　[7：0]
偏移量：　　　0x03
复位值：　　　0x00

位域	位域名称	位宽	访问	描述
7:0	RXR	8	R	存放最后一个接收到的字节

3.8.6 命令控制寄存器（CR）

中文名：　　　命令寄存器
寄存器位宽：　[7：0]
偏移量：　　　0x04
复位值：　　　0x00

位域	位域名称	位宽	访问	描述
7	STA	1	W	产生 START 信号
6	STO	1	W	产生 STOP 信号
5	RD	1	W	产生读信号
4	WR	1	W	产生写信号
3	ACK	1	W	产生应答信号
2:1	Reserved	2	W	保留

0	IACK	1	W	产生中断应答信号

都是在 I2C 发送数据后硬件自动清零。对这些位读操作时候总是读回‘0’。

3.8.7 状态寄存器（SR）

中文名： 状态寄存器
寄存器位宽： [7：0]
偏移量： 0x04
复位值： 0x00

位域	位域名称	位宽	访问	描述
7	RxACK	1	R	收到应答位 1 没收到应答位 0 收到应答位
6	Busy	1	R	I2c 总线忙标志位 1 总线在忙 0 总线空闲
5	AL	1	R	当 I2C 核失去 I2C 总线控制权时，该位置 1
4:2	Reserved	3	R	保留
1	TIP	1	R	指示传输的过程 1 表示正在传输数据 0 表示数据传输完毕
0	IF	1	R	中断标志位，一个数据传输完，或另外一个器件发起数据传输，该位置 1

3.9 LPB MISC 寄存器

本模块寄存器物理地址基址为 0x1F004100。

3.9.1 GPIO_OE_7_0

中文名： GPIO[7:0] 输出使能寄存器
寄存器位宽： [7：0]
偏移量： 0x00
复位值： 0x00

位域	位域名称	位宽	访问	描述
7:0	GPIO_OE_7_0	8	R/W	GPIO[7:0]输出使能寄存器, 1 –有效

3.9.2 GPIO_OE_15_8

中文名： GPIO[15:8] 输出使能寄存器
寄存器位宽： [7：0]
偏移量： 0x01
复位值： 0x00

位域	位域名称	位宽	访问	描述

7:0	GPIO_OE_15_8	8	R/W	GPIO[15:8]输出使能寄存器, 1 –有效

3.9.3 GPIO_OE_23_16

中文名： GPIO[23:16] 输出使能寄存器
寄存器位宽： [7：0]
偏移量： 0x02
复位值： 0x00

位域	位域名称	位宽	访问	描述
7:0	GPIO_OE_23_16	8	R/W	GPIO[23:16]输出使能寄存器, 1 –有效

3.9.4 GPIO_OE_28_24

中文名： GPIO[28:24] 输出使能寄存器
寄存器位宽： [7：0]
偏移量： 0x03
复位值： 0x00

位域	位域名称	位宽	访问	描述
7:5	Reserved	3		保留
4:0	GPIO_OE_28_24	5	R/W	GPIO[28:24]输出使能寄存器, 1 –有效

3.9.5 GPIO_I_7_0

中文名： GPIO[7:0] 引脚输入寄存器
寄存器位宽： [7：0]
偏移量： 0x10
复位值： 0x00

位域	位域名称	位宽	访问	描述
7:0	GPIO_I_7_0	8	R/-	GPIO[7:0]引脚输入寄存器

3.9.6 GPIO_I_15_8

中文名： GPIO[15:8] 引脚输入寄存器
寄存器位宽： [7：0]
偏移量： 0x11
复位值： 0x00

位域	位域名称	位宽	访问	描述
7:0	GPIO_I_15_8	8	R/-	GPIO[15:8]引脚输入寄存器

3.9.7 GPIO_I_23_16

中文名： GPIO[23:16] 引脚输入寄存器
寄存器位宽： [7：0]
偏移量： 0x12
复位值： 0x00

位域	位域名称	位宽	访问	描述

7:0	GPIO_I_23_16	8	R/-	GPIO[23:16]引脚输入寄存器

3.9.8 GPIO_I_28_24

中文名： GPIO[28:24] 引脚输入寄存器
寄存器位宽： [7：0]
偏移量： 0x13
复位值： 0x00

位域	位域名称	位宽	访问	描述
7:5	Reserved	3		保留
4:0	GPIO_I_28_24	5	R/-	GPIO[28:24]引脚输入寄存器

3.9.9 GPIO_O_7_0

中文名： GPIO[7:0] 引脚输出寄存器
寄存器位宽： [7：0]
偏移量： 0x20
复位值： 0x00

位域	位域名称	位宽	访问	描述
7:0	GPIO_O_7_0	8	R/W	GPIO[7:0]引脚输出寄存器

3.9.10 GPIO_O_15_8

中文名： GPIO[15:8] 引脚输出寄存器
寄存器位宽： [7：0]
偏移量： 0x21
复位值： 0x00

位域	位域名称	位宽	访问	描述
7:0	GPIO_O_15_8	8	R/W	GPIO[15:8]引脚输出寄存器

3.9.11 GPIO_O_23_16

中文名： GPIO[23:16] 引脚输出寄存器
寄存器位宽： [7：0]
偏移量： 0x22
复位值： 0x00

位域	位域名称	位宽	访问	描述
7:0	GPIO_O_23_16	8	R/W	GPIO[23:16]引脚输出寄存器

3.9.12 GPIO_O_28_24

中文名： GPIO[28:24] 引脚输出寄存器
寄存器位宽： [7：0]
偏移量： 0x23
复位值： 0x00

位域	位域名称	位宽	访问	描述
7:5	Reserved	3		保留

4:0	GPIO_O_28_24	5	R/W	GPIO[28:24]引脚输出寄存器

3.9.13 LPB_MISC_CFG

中文名： LPB MISC 配置寄存器
寄存器位宽： [7：0]
偏移量： 0x40
复位值： 0x7F

位域	位域名称	位宽	访问	描述
7	Reserved	1		保留
6	LED_EN	1	R/W	在使用 PHY LED 时，置 1
5	AC97_EN	1	R/W	在使用 AC97 时，置 1
4	URT1_EN	1	R/W	在使用 UART ＃1 时，置 1
3	URT0_EN	1	R/W	在使用 UART ＃0 时，置 1
2	M_EN	1	R/W	在使用 PS/2 Mouse 时，置 1
1	K_EN	1	R/W	在使用 PS/2 Keyboard 时，置 1
0	SPI_EN	1	R/W	在使用 SPI 时，置 1

3.10 AC97 控制器寄存器

本模块寄存器物理地址基址为 0x1F004200。

寄存器名	宽度	偏移量	描述
CSR	2	0x00	配置状态寄存器
OCC0	24	0x04	输出通道配置寄存器 0
OCC1	24	0x08	保留
OCC2	24	0x0c	保留
ICC	24	0x10	输入通道配置寄存器
CODEC_ID	32	0x14	Codec ID 寄存器
CRAC	32	0x18	Codec 寄存器访问命令
OC0	20	0x20	输出声道 0
OC1	20	0x24	输出声道 1
OC2	20	0x28	保留
OC3	20	0x2c	保留
OC4	20	0x30	保留
OC5	20	0x34	保留
OC6	20	0x38	保留
OC7	20	0x3c	保留
OC8	20	0x40	保留
IC0	20	0x44	保留
IC1	20	0x48	保留
IC2	20	0x4c	输入声道 2
INTRAW	32	0x54	中断状态寄存器
INTM	32	0x58	中断掩膜

寄存器名	宽度	偏移量	描述
INTS	32	0x5c	保留

3.10.1 CSR 寄存器

中文名： 配置状态寄存器
寄存器位宽： [31：0]
偏移量： 0x00
复位值： 0x00000000

位域	位域名称	位宽	访问	描述
31:2	Reserved	30	R	保留
1	RESUME	1	R/W	挂起 读此位返回现在 AC97 子系统的状态 1：AC97 子系统挂起 0：正常工作状态 在挂起状态下，写入 1 到该位，将会开始恢复操作。
0	RST_FORCE	1	W	AC97 冷启动 写入 1 会导致 AC97 Codec 冷启动

3.10.2 OCC 寄存器 0

中文名： 输出通道配置寄存器
寄存器位宽： [31：0]
偏移量： 0x04
复位值： 0x00004141

位域	位域名称	位宽	访问	描述
31:24	Reserved	8	R/W	保留
23:16	Reserved	8	R/W	保留
15:8	OC1_CFG_R	8	R/W	输出通道 1：右声道配置。 详见 3.10.4 节
7:0	OC0_CFG_L	8	R/W	输出通道 0：左声道配置。 详见 3.10.4 节

3.10.3 ICC 寄存器

中文名： 输入通道配置寄存器
寄存器位宽： [31：0]
偏移量： 0x10
复位值： 0x00410000

位域	位域名称	位宽	访问	描述
31:24	Reserved	8	R/W	保留
23:16	IC_CFG_MIC	8	R/W	输入通道 2：MIC 声道配置。 详见 3.10.4 节
15:8	Reserved	8	R/W	保留
7:0	Reserved	8	R/W	保留

3.10.4 （输入输出）通道寄存器配置

位域	位域名称	位宽	访问	描述
7	Reserved	1	R/W	保留
6	DMA_EN	1	R/W	DMA 使能 1：DMA 打开 0：DMA 关闭
5:4	FIFO_THRES	2	R/W	FIFO 门限 5：4 输出通道 输入通道 00 FIFO 1/4 空 FIFO 1/4 满 01 FIFO 1/2 空 FIFO 1/2 满 10 FIFO 3/4 空 FIFO 3/4 满 11 FIFO 全空 FIFO 全满
3:2	SW	2	R/W	采样位数 00：8 位 01：16 位
1	VSR	1	R/W	采样率 1：采样率可变 0：采样率固定（48KHz）
0	CH_EN	1	R/W	通道使能 1：通道打开 0：通道关闭（或者进入节能状态）

3.10.5 Codec 寄存器访问命令

中文名：　Codec 寄存器访问命令
寄存器位宽：　[31：0]
偏移量：　0x18
复位值：　0x00000000

位域	位域名称	位宽	访问	描述
31	CODEC_WR	1	R/W	读/写选择 1：读 0：写
30:23	Reserved	8	R	保留
22:16	CODEC_ADR	7	R/W	Codec 寄存器地址
15:0	CODEC_DAT	16	R/W	Codec 寄存器数据

3.10.6 中断掩膜寄存器/中断状态寄存器

中文名：　中断状态/中断掩膜寄存器
寄存器位宽：　[31：0]
偏移量：　0x54/58
复位值：　0x00000000

位域	位域名称	位宽	访问	描述
31	IC_FULL	1	R/RW	输入通道 2：FIFO 满

30	IC_TH_INT	1	R/RW	输入通道 2：FIFO 达到门限
29:8	Reserved	22	R/RW	保留
7	OC1_FULL	1	R/RW	输出通道 1：FIFO 满
6	OC1_EMPTY	1	R/RW	输出通道 1：FIFO 空
5	OC1_ TH_INT	1	R/RW	输出通道 1：FIFO 达到门限
4	OC0_FULL	1	R/RW	输出通道 0：FIFO 满
3	OC0_EMPTY	1	R/RW	输出通道 0：FIFO 空
2	OC0_ TH_INT	1	R/RW	输出通道 0：FIFO 达到门限
1	CW_DONE	1	R/RW	Codec 寄存器写完成
0	CR_DONE	1	R/RW	Codec 寄存器读完成

3.11 AC97 DMA 控制器寄存器

本模块寄存器物理地址基址为 0x1F004280。

3.11.1 OUT_ADDR

中文名： 输出命令 DMA 地址寄存器
寄存器位宽： [31：0]
偏移量： 0x00
复位值： 0x00000000

位域	位域名称	位宽	访问	描述
31:28	Reserved	4		保留
27:2	OUT_ADDR	26	-/W	声音输出通道命令 FIFO DMA 地址寄存器
1:0	Reserved	2		保留

3.11.2 OUT_LENGTH

中文名： 输出命令长度寄存器
寄存器位宽： [31：0]
偏移量： 0x04
复位值： 0x00000000

位域	位域名称	位宽	访问	描述
31:20	Reserved	12		保留
19:0	OUT_LENGTH	20	-/W	声音输出通道命令 FIFO DMA 长度寄存器

3.11.3 OFIFO_RPT

中文名： 输出命令 FIFO 当前读指针
寄存器位宽： [31：0]
偏移量： 0x08
复位值： 0x00000000

位域	位域名称	位宽	访问	描述
31:1	Reserved	31		保留
0:0	OFIFO_RPT	1	R/-	声音输出通道命令 FIFO 当前读指针

3.11.4 OFIFO_WPT

中文名： 输出命令 FIFO 当前读指针
寄存器位宽： [31：0]
偏移量： 0x0C
复位值： 0x00000000

位域	位域名称	位宽	访问	描述
31:1	Reserved	31		保留
0:0	OFIFO_WPT	1	R/-	声音输出通道命令 FIFO 当前写指针

3.11.5 IN_ADDR

中文名： 输入命令 DMA 地址寄存器
寄存器位宽： [31：0]
偏移量： 0x10
复位值： 0x00000000

位域	位域名称	位宽	访问	描述
31:28	Reserved	4		保留
27:2	IN_ADDR	26	-/W	声音输入通道命令 FIFO DMA 地址寄存器
1:0	Reserved	3		保留

3.11.6 IN_LENGTH

中文名： 输入命令长度寄存器
寄存器位宽： [31：0]
偏移量： 0x14
复位值： 0x00000000

位域	位域名称	位宽	访问	描述
31:20	Reserved	12		保留
19:0	IN_LENGTH	20	-/W	声音输入通道命令 FIFO DMA 长度寄存器

3.11.7 IFIFO_RPT

中文名： 输入命令 FIFO 当前读指针
寄存器位宽： [31：0]
偏移量： 0x18
复位值： 0x00000000

位域	位域名称	位宽	访问	描述
31:1	Reserved	31		保留
0:0	IFIFO_RPT	1	R/-	声音输入通道命令 FIFO 当前读指针

3.11.8 IFIFO_WPT

中文名： 输入命令 FIFO 当前读指针
寄存器位宽： [31：0]
偏移量： 0x1C
复位值： 0x00000000

位域	位域名称	位宽	访问	描述
31:1	Reserved	31		保留
0:0	IFIFO_WPT	1	R/-	声音输入通道命令 FIFO 当前写指针

3.11.9 OUT_MODE

中文名： 输出命令工作模式寄存器
寄存器位宽： [31：0]
偏移量： 0x20
复位值： 0x00010001

位域	位域名称	位宽	访问	描述
31:17	Reserved	15		保留
16:16	OUT_MODE	1	R/W	声音输出通道 PCM 工作方式寄存器 00: 8 bit ; 01: 16 bit ; 10: 32 bit
15:0	Reserved	16		保留

3.11.10 INT_MASK

中文名： 中断屏蔽寄存器
寄存器位宽： [31：0]
偏移量： 0x28
复位值： 0x0000003F

位域	位域名称	位宽	访问	描述
31:16	Reserved	26		保留
5:0	INT_MASK	6	R/W	中断屏蔽寄存器 5： 声音输入通道命令 FIFO 满 4： 声音输入通道命令 FIFO 空 3： 声音输入通道一条 DMA 命令完成 2： 声音输出通道命令 FIFO 满 1： 声音输出通道命令 FIFO 空 0： 声音输出通道一条 DMA 命令完成

3.11.11 INT_CAUSE

中文名： 中断原因及响应寄存器
寄存器位宽： [31：0]
偏移量： 0x2C
复位值： 0x00000012

位域	位域名称	位宽	访问	描述

位域	位域名称	位宽	访问	描述
31:16	Reserved	26		保留
5:0	INT_CAUSE	6	R/W	中断原因及响应寄存器 5:　声音输入通道命令 FIFO 满 4:　声音输入通道命令 FIFO 空 3:　声音输入通道一条 DMA 命令完成 2:　声音输出通道命令 FIFO 满 1:　声音输出通道命令 FIFO 空 0:　声音输出通道一条 DMA 命令完成 注：只有 BIT 0 与 BIT 3 是可以响应的，其余位为 FIFO 状态位，只能用 INT_MASK 寄存器屏蔽。

3.12 MAC 控制器寄存器

本模块包括两个功能完全相同，能够同时工作的 MAC 模块；他们寄存器完全一样，所不同的是他们的物理地址基址。他们的地址基址分别为 ：

MAC #0 的物理地址基址为 0x1F005200；

MAC #1 的物理地址基址为 0x1F005300。

3.12.1 MAC0_BUS_MODE

中文名：　网口 0 总线模式寄存器

寄存器位宽：　[31：0]

偏移量：　0x00

复位值：　0xFE000000

位域	位域名称	位宽	访问	描述
20	DESBLE	1	R/W	0:数据描述符为小尾端模式 1:数据描述符为大尾端模式
19:17	TXAP	3	R/W	发送自动查询，000:无操作 其它:开始发送查询
13:8	DMABL	6	R/W	DMA 一次传输最大长度设置
7	DATABLE	1	R/W	0:数据缓存为小尾端模式 1:数据缓存为大尾端模式
6:2	DSL	5	R/W	在描述符采用环式组织模式时，描述符之间间隔（单位：字）
1	PRI	1	R/W	0:接收相对发送具有较高优先级 1:发送和接收具有相同优先级
0	SWRESET	1	R/W	软件复位(除 HSB 接口部分)

3.12.2 MAC0_TX_POLL_REQ

中文名：　网口 0 发送查询请求寄存器

寄存器位宽：　[31：0]

偏移量：　0x08

复位值：　0xFFFFFFFF

位域	位域名称	位宽	访问	描述

31:0	TXCMD	32	W	写入任何非 0 值使得 MAC 检查是否有帧需要发送

3.12.3 MAC0_RX_POLL_REQ

中文名： 网口 0 接收查询请求寄存器
寄存器位宽： [31：0]
偏移量： 0x10
复位值： 0xFFFFFFFF

位域	位域名称	位宽	访问	描述
31:0	RXCMD	32	W	写入任何非 0 值使得 MAC 检查是否有帧需要接收

3.12.4 MAC0_RX_LIST_BASE_ADDR

中文名： 网口 0 接收描述符地址寄存器
寄存器位宽： [31：0]
偏移量： 0x18
复位值： 0xFFFFFFFF

位域	位域名称	位宽	访问	描述
31:0	RXBASE	32	R/W	接收帧描述符链表的起始地址

3.12.5 MAC0_TX_LIST_BASE_ADDR

中文名： 网口 0 发送描述符地址寄存器
寄存器位宽： [31：0]
偏移量： 0x20
复位值： 0xFFFFFFFF

位域	位域名称	位宽	访问	描述
31:0	TXBASE	32	R/W	发送帧描述符链表的起始地址

3.12.6 MAC0_STATUS

中文名： 网口 0 状态寄存器
寄存器位宽： [31：0]
偏移量： 0x28
复位值： 0xF0000000

位域	位域名称	位宽	访问	描述
22:20	TXST	3	RO	发送过程状态
19:17	RXST	3	RO	接收过程状态
16	NINT	1	R/W	正常中断（写入 1 清除该中断）
15	AINT	1	R/W	异常中断（写入 1 清除该中断）
14	ERXI	1	R/W	接收早期中断（写入 1 清除该中断）
11	GTE	1	R/W	通用计数器超时中断(写入 1 清除该中断）
10	ETXI	1	R/W	发送早期中断（写入 1 清除该中断）
8	RPS	1	R/W	接收过程停止中断（写入 1 清除该中断）
7	RXBU	1	R/W	接收缓存不可用中断(写入 1 清除该中断）
6	RXI	1	R/W	帧接收完成中断（写入 1 清除该中断）
5	UNF	1	R/W	发送下溢中断（写入 1 清除该中断）

位域	位域名称	位宽	访问	描述
2	TXBU	1	R/W	发送缓存不可用中断(写入 1 清除该中断)
1	TPS	1	R/W	发送过程停止中断（写入 1 清除该中断）
0	TXI	1	R/W	帧发送完成（写入 1 清除该中断）

3.12.7 MAC0_OPERATION_MODE

中文名：　　　网口 0 操作模式寄存器
寄存器位宽：　[31：0]
偏移量：　　　0x30
复位值：　　　0x32000040

位域	位域名称	位宽	访问	描述
30	RALL	1	R/W	0:进行地址过滤 1:接收所有类型的帧
22	TTM	1	R/W	1:发送带宽阈值为 10Mb/s 0:发送带宽阈值为 100Mb/s
21	SF	1	R/W	0:发送 FIFO 不采用存储转发模式 1:发送 FIFO 采用存储转发模式
15:14	TRC	2	R/W	发送队列阈值控制字
13	TXS	1	R/W	0:停止帧发 1:开始帧发送送
9	FD	1	R/W	0:半双工模式 1:全双工模式
7	PMC	1	R/W	0:多播地址被地址检查模块过滤 1:多播地址不被过滤
6	PR	1	R/W	0:正常模式 1:混杂模式
4	IFM	1	R/W	0:正常模式 1:地址过滤为反向过滤模式（只读，由配置帧的状态决定）
3	PBF	1	R/W	0:正常模式 1:允许坏帧被接收
2	HO	1	R/W	0:正常模式 1:采用 hash 地址过滤模式(只读，由配置帧的状态决定）
1	RXS	1	R/W	0:停止帧接收 1:开始帧接收
0	AFM	1	R/W	0:根据配置帧的物理地址进行地址过滤 1:根据配置帧的多播地址进行地址过滤

3.12.8 MAC0_INTERRUPT_EN

中文名：　　　网口 0 中断使能寄存器
寄存器位宽：　[31：0]
偏移量：　　　0x38
复位值：　　　0xF3FE0000

位域	位域名称	位宽	访问	描述
16	NINTE	1	R/W	0:正常中断不使能 1:正常中断使能
15	AINTE	1	R/W	0:非正常中断不使能 1:非正常中断使能
14	ERXIE	1	R/W	0:接收早期中断不使能 1:接收早期中断使能
11	GTE	1	R/W	0:通用时钟超时中断不使能 1:通用时钟超时中断使能

位域	位域名称	位宽	访问	描述
10	ETXIE	1	R/W	0:发送早期中断不使能 1:发送早期中断使能
8	RPSE	1	R/W	0:接收停止中断不使能 1:接收停止中断使能
7	RXBUE	1	R/W	0:接收缓存不可用中断不使能 1:接收缓存不可用中断使能
6	RXIE	1	R/W	0:接收中断不使能 1:接收中断使能
5	UNFE	1	R/W	0:下溢中断不使能 1:下溢中断使能
2	TXBUE	1	R/W	0:发送缓存不可用中断不使能 1:发送缓存不可用中断使能
1	TPSE	1	R/W	0:发送停止中断不使能 1:发送停止中断使能
0	TXIE	1	R/W	0:发送中断不使能 1:发送中断使能

3.12.9 MAC0_MISSED_FRAME_STATISTIC

中文名： 网口 0 帧丢失统计寄存器

寄存器位宽： [31：0]

偏移量： 0x40

复位值： 0xE0000000

位域	位域名称	位宽	访问	描述
28	OCO	1	RO	0:溢出计数器正常 1:溢出计数器溢出
27:17	FOVC	11	RO	接收 FIFO 满而未被接收帧的个数（读清）
16	MFCO	1	RO	0:丢失帧记数器正常 1:丢失帧计数器溢出
15:0	MFC	16	RO	由于接收描述符不可用而未被丢弃帧的个数（读清）

3.12.10 MAC0_SMI_EEPROM_CTL

中文名： 网口 0SMI 与 EEPROM 控制寄存器

寄存器位宽： [31：0]

偏移量： 0x48

复位值： 0xFFF483FB

位域	位域名称	位宽	访问	描述
19	MDI	1	RO	SMI 数据输入
18	MII	1	R/W	0:SMI 读操作 1:SMI 写操作
17	MDO	1	R/W	SMI 数据输出
16	MDC	1	R/W	SMI 时钟输出

3.12.11 MAC0_BYTE_ALIGN

中文名： 网口 0 字节对齐使能寄存器

寄存器位宽： [31：0]

偏移量： 0x50

复位值： 0x00000000

位域	位域名称	位宽	访问	描述

0	INSERT_EN	4	R/W	1: 在接收帧的头部加入两空白字节后再存入内存，使得帧的数据部分在内存中字对齐； 0: 关闭该功能。

3.12.12 MAC0_GPT_IM_CTL

中文名：　　网口 0 定时器与中断缓和寄存器
寄存器位宽：　[31：0]
偏移量：　　0x58
复位值：　　0xFFFE0000

位域	位域名称	位宽	访问	描述
31	TIMU	1	R/W	设置发送接收定时器的时间单位 0: 1.024uS 1: 16.38uS
30:27	TXTIM	4	R/W	发送定时器，定义发送一帧到发出发送中断之间的时间间隔（以 TIMU 为单位)
26:24	NTXP	3	R/W	定义发送多少帧后才发出发送中断
23:20	RXTIM	4	R/W	接收定时器，定义接收一帧到发出发送中断之间的时间间隔（以 TIMU 为单位)
19:17	NRXP	3	R/W	定义接收多少帧后才发出发送中断
16	CON	1	R/W	0:通用定时器工作在 one-shot 模式 1:通用定时器工作在连续模式
15:0	TIMER	16	R/W	定时器记数值（单位为:16.384uS）

3.12.13 MAC1_BUS_MODE

中文名：　　网口 1 总线模式寄存器
寄存器位宽：　[31：0]
偏移量：　　0x00
复位值：　　0xFE000000

位域	位域名称	位宽	访问	描述
20	DESBLE	1	R/W	0:数据描述符为小尾端模式 1:数据描述符为大尾端模式
19:17	TXAP	3	R/W	发送自动查询，000:无操作 其它:开始发送查询
13:8	DMABL	6	R/W	DMA 一次传输最大长度设置
7	DATABLE	1	R/W	0:数据缓存为小尾端模式 1:数据缓存为大尾端模式
6:2	DSL	5	R/W	在描述符采用环式组织模式时，描述符之间间隔（单位：字）
1	PRI	1	R/W	0:接收相对发送具有较高优先级 1:发送和接收具有相同优先级
0	SWRESET	0	R/W	软件复位(除 HSB 接口部分)

3.12.14 MAC1_TX_POLL_REQ

中文名：　　网口 1 发送查询请求寄存器
寄存器位宽：　[31：0]

偏移量：　0x08
复位值：　0xFFFFFFFF

位域	位域名称	位宽	访问	描述
31:0	TXCMD	32	W	写入任何非 0 值使得 MAC 检查是否有帧需要发送

3.12.15 MAC1_RX_POLL_REQ

中文名：　网口 1 接收查询请求寄存器
寄存器位宽：　[31：0]
偏移量：　0x10
复位值：　0xFFFFFFFF

位域	位域名称	位宽	访问	描述
31:0	RXCMD	32	W	写入任何非 0 值使得 MAC 检查是否有帧需要接收

3.12.16 MAC1_RX_LIST_BASE_ADDR

中文名：　网口 1 接收描述符地址寄存器
寄存器位宽：　[31：0]
偏移量：　0x18
复位值：　0xFFFFFFFF

位域	位域名称	位宽	访问	描述
31:0	RXBASE	32	R/W	接收帧描述符链表的起始地址

3.12.17 MAC1_TX_LIST_BASE_ADDR

中文名：　网口 1 发送描述符地址寄存器
寄存器位宽：　[31：0]
偏移量：　0x20
复位值：　0xFFFFFFFF

位域	位域名称	位宽	访问	描述
31:0	TXBASE	32	R/W	发送帧描述符链表的起始地址

3.12.18 MAC1_STATUS

中文名：　网口 1 状态寄存器
寄存器位宽：　[31：0]
偏移量：　0x28
复位值：　0xF0000000

位域	位域名称	位宽	访问	描述
22:20	TXST	3	RO	发送过程状态
19:17	RXST	3	RO	接收过程状态
16	NINT	1	R/W	正常中断（写入 1 清除该中断）
15	AINT	1	R/W	异常中断（写入 1 清除该中断）
14	ERXI	1	R/W	接收早期中断（写入 1 清除该中断）
11	GTE	1	R/W	通用计数器超时中断(写入 1 清除该中断）
10	ETXI	1	R/W	发送早期中断（写入 1 清除该中断）
8	RPS	1	R/W	接收过程停止中断（写入 1 清除该中断）

位域	位域名称	位宽	访问	描述
7	RXBU	1	R/W	接收缓存不可用中断(写入 1 清除该中断)
6	RXI	1	R/W	帧接收完成中断（写入 1 清除该中断）
5	UNF	1	R/W	发送下溢中断（写入 1 清除该中断）
2	TXBU	1	R/W	发送缓存不可用中断(写入 1 清除该中断)
1	TPS	1	R/W	发送过程停止中断（写入 1 清除该中断）
0	TXI	1	R/W	帧发送完成（写入 1 清除该中断）

3.12.19 MAC1_OPERATION_MODE

中文名：　　网口 1 操作模式寄存器
寄存器位宽：　[31：0]
偏移量：　　0x30
复位值：　　0x32000040

位域	位域名称	位宽	访问	描述
30	RALL	1	R/W	0:进行地址过滤 1:接收所有类型的帧
22	TTM	1	R/W	1:发送带宽阈值为 10Mb/s 0:发送带宽阈值为 100Mb/s
21	SF	1	R/W	0:发送 FIFO 不采用存储转发模式 1:发送 FIFO 采用存储转发模式
15:14	TRC	2	R/W	发送队列阈值控制字
13	TXS	1	R/W	0:停止帧发 1:开始帧发送送
9	FD	1	R/W	0:半双工模式 1:全双工模式
7	PMC	1	R/W	0:多播地址被地址检查模块过滤 1:多播地址不被过滤
6	PR	1	R/W	0:正常模式 1:混杂模式
4	IFM	1	R/W	0:正常模式 1:地址过滤为反向过滤模式（只读，由配置帧的状态决定）
3	PBF	1	R/W	0:正常模式 1:允许坏帧被接收
2	HO	1	R/W	0:正常模式 1:采用 hash 地址过滤模式(只读，由配置帧的状态决定)
1	RXS	1	R/W	0:停止帧接收 1:开始帧接收
0	AFM	1	R/W	0:根据配置帧的物理地址进行地址过滤 1:根据配置帧的多播地址进行地址过滤

3.12.20 MAC1_INTERRUPT_EN

中文名：　　网口 1 中断使能寄存器
寄存器位宽：　[31：0]
偏移量：　　0x38
复位值：　　0xF3FE0000

位域	位域名称	位宽	访问	描述
16	NINTE	1	R/W	0:正常中断不使能 1:正常中断使能
15	AINTE	1	R/W	0:非正常中断不使能 1:非正常中断使能
14	ERXIE	1	R/W	0:接收早期中断不使能 1:接收早期中断使能

位域	位域名称	位宽	访问	描述
11	GTE	1	R/W	0:通用时钟超时中断不使能 1:通用时钟超时中断不使能
10	ETXIE	1	R/W	0:发送早期中断不使能 1:发送早期中断使能
8	RPSE	1	R/W	0:接收停止中断不使能 1:接收停止中断使能
7	RXBUE	1	R/W	0:接收缓存不可用中断不使能 1:接收缓存不可用中断使能
6	RXIE	1	R/W	0:接收中断不使能 1:接收中断使能
5	UNFE	1	R/W	0:下溢中断不使能 1:下溢中断使能
2	TXBUE	1	R/W	0:发送缓存不可用中断不使能 1:发送缓存不可用中断使能
1	TPSE	1	R/W	0:发送停止中断不使能 1:发送停止中断使能
0	TXIE	1	R/W	0:发送中断不使能 1:发送中断使能

3.12.21 MAC1_MISSED_FRAME_STATISTIC

中文名：　　网口 1 帧丢失统计寄存器
寄存器位宽：[31：0]
偏移量：　　0x40
复位值：　　0xE0000000

位域	位域名称	位宽	访问	描述
28	OCO	1	RO	0:溢出计数器正常 1:溢出计数器溢出
27:17	FOVC	11	RO	接收 FIFO 满而未被接收帧的个数（读清）
16	MFCO	1	RO	0:丢失帧记数器正常 1:丢失帧计数器溢出
15:0	MFC	16	RO	由于接收描述符不可用而未被丢弃帧的个数（读清）

3.12.22 MAC1_SMI_EEPROM_CTL

中文名：　　网口 1SMI 与 EEPROM 控制寄存器
寄存器位宽：[31：0]
偏移量：　　0x48
复位值：　　0xFFF483FB

位域	位域名称	位宽	访问	描述
19	MDI	1	RO	SMI 数据输入
18	MII	1	R/W	0:SMI 读操作 1:SMI 写操作
17	MDO	1	R/W	SMI 数据输出
16	MDC	1	R/W	SMI 时钟输出

3.12.23 MAC1_BYTE_ALIGN

中文名：　　网口 1 字节对齐使能寄存器
寄存器位宽：[31：0]
偏移量：　　0x50
复位值：　　0x00000000

位域	位域名称	位宽	访问	描述
0	INSERT_EN	4	R/W	1: 在接收帧的头部加入两空白字节后再存入内存，使得帧的数据部分在内存中字对齐； 0: 关闭该功能。

3.12.24 MAC1_GPT_IM_CTL

中文名： 网口 1 定时器与中断缓和寄存器

寄存器位宽： [31：0]

偏移量： 0x58

复位值： 0xFFFE0000

位域	位域名称	位宽	访问	描述
31	TIMU	1	R/W	设置发送接收定时器的时间单位 0: 1.024uS 1: 16.38uS
30:27	TXTIM	4	R/W	发送定时器，定义发送一帧到发出发送中断之间的时间间隔（以 TIMU 为单位)
26:24	NTXP	3	R/W	定义发送多少帧后才发出发送中断
23:20	RXTIM	4	R/W	接收定时器，定义接收一帧到发出发送中断之间的时间间隔（以 TIMU 为单位)
19:17	NRXP	3	R/W	定义接收多少帧后才发出发送中断
16	CON	1	R/W	0:通用定时器工作在 one-shot 模式 1:通用定时器工作在连续模式
15:0	TIMER	16	R/W	定时器记数值（单位为:16.384uS）

4 芯片功能

SOC3210 系列芯片是一款集成了中央处理器、外部存储控制器、以太网交换功能、周边元件扩展接口、音频控制器、通用输入/输出控制器、中断控制器、串行外围设备控制器、串行通讯总线控制器等丰富功能的 SOC 芯片。

首先，SOC3210 系列芯片集成了通用处理器 32 位支持 MIPS32 指令集的 RISC CPU，可以适应控制、计算等多种应用。其次，SOC3210 系列芯片的外部储存控制器 EMI 能够支持多种存储芯片，包括 SDRAM、EEPROM、FLASH 等。SOC3210 系列芯片还集成了一个液晶显示控制器，支持 STN/TFT 液晶面板，最大分辨率 800×600，最大支持 65536 色。SOC3210 系列芯片集成了 AC97 音频控制器，支持麦克风输入和 2 频道立体声输出，支持 16、18 和 20 位采样精度。SOC3210 系列芯片集成了两个 MAC 模块，提供两路 MII 接口，支持 10M/100M 自适应的网络。SOC3210 系列芯片集成了键盘/鼠标控制器，可以连接键盘/鼠标进行使用。SOC3210 系列芯片还集成了 SPI、I^2C 等总线控制器，可以连接各种各样的外围设备。SOC3210 系列芯片还集成了中断控制器、WatchDog、GPIO 等功能。

本章以下各节将对 SOC3210 系列芯片内的丰富功能详细介绍。

4.1 32 位 RISC CPU

32 位 RISC CPU 核支持 MIP32 指令集中的 32 位定点指令，包括定点运算指令、转移指令、访存指令，以及控制指令等，能够兼顾通用和嵌入式应用需求。32 位 RISC CPU 的工作频率为 160~266MHz，采用 5 级流水线，32 位整数单元和非阻塞 CACHE 的存储管理单元。通过巧妙地把数据运算、转移控制以及例外处理紧密地融合在一起，32 位 RISC CPU 流水线设计做到了简洁流畅。32 位 RISC CPU 的微体系结构如图 4-1 所示。下面按照流水线的顺序介绍指令在 32 位 RISC CPU 的处理过程。

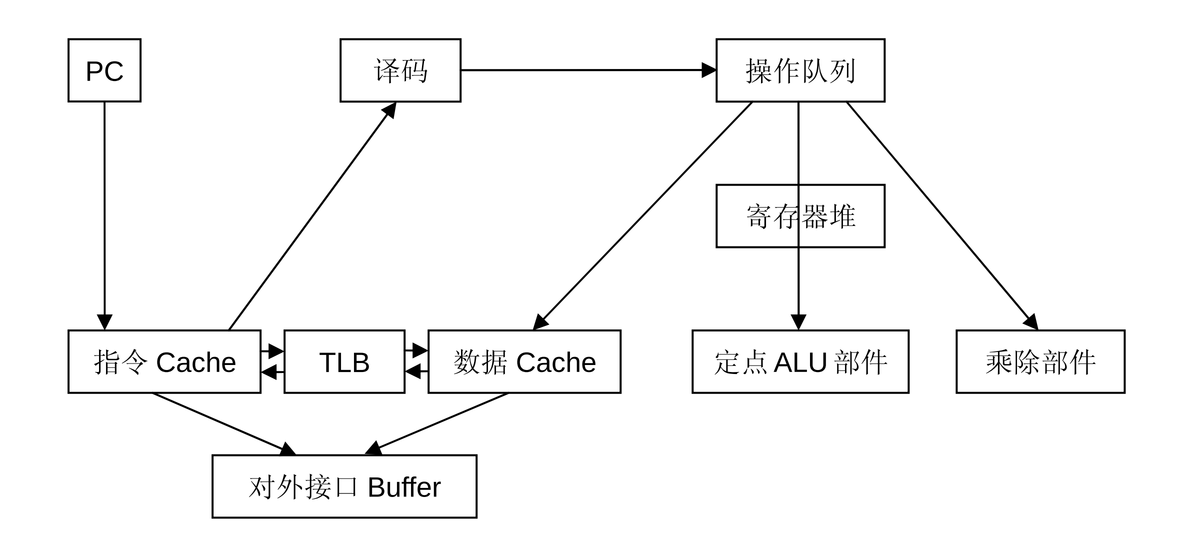


图 4-1 32 位 RISC CPU 微体系结构图

1) 取指

32 位 RISC CPU 的指令执行从取指开始。根据 PC 的值，访问 TLB 将虚地址转换成物理地址，同时访问指令 CACHE，读取相应 CACHE 行的 TAG 和指令，将 TAG 和转换后的物理地址高位比较，选择正确的指令送往译码部件。

PC

PC 在系统 RESET 后，初始值为 0xbfc0_0000。每次成功取指则 PC 加 4。当发生异常时，PC 将指向异常入口地址。当译码出条件转移指令时，PC 将指向转移跳转的目标地址。当发生转移预测错误时，PC 将指向转移指令正确的目标地址。

TLB

32 位 RISC CPU 的 TLB 有 32 项，完成虚地址到物理地址的转换，可支持最大 4GB 的物理地址空间。虚页的大小可以设置，其大小可以为 4KB~16MB（以 4 倍倍增）。为了提高安全性，32 位 RISC CPU 支持三种操作模式：用户模式、管理模式和核心操作模式。处理器的当前工作模式取决于系统状态寄存器中的控制位。32 位 RISC CPU 提供了一种新的安全机制，可有效防治缓冲区溢出攻击。缓冲区溢出是一种普遍存在的安全漏洞，容易被计算机病毒和网络攻击者所利用。

指令 CACHE

32 位 RISC CPU 的指令 CACHE 一共 16KB 大小，四路组相联，每路 4KB 大小，每个 CACHE 行大小为 32B。当发生 CACHE 不命中时，从内存中读回对应的 CACHE 行，然后随机替换四路中的一路，将读回的 CACHE 行填入指

令 CACHE 中。

2) 指令译码

取来的指令经过译码部件，翻译成 32 位 RISC CPU 的内部操作。这主要是将原先不完全相同格式的指令，翻译成完全相同的规整格式的内部操作，方便内部的各个部分识别处理相应的操作。当译码时遇到条件转移指令，32 位 RISC CPU 进行静态转移预测，即预测所有的条件转移指令将发生跳转，并且计算出跳转的目标地址，修改 PC 的值。

译码好的内部操作将送往操作队列，在进入操作队列的同时，查找之前的操作是否与当前操作存在数据相关，如果存在数据相关，则将原操作数指向操作队列中对应的操作，并记录相应信息，此过程是为了将来的乱序执行的重命名。

操作队列

32 位 RISC CPU 有 8 项操作队列。操作队列用来保存译码后的内部操作，从译码之后到指令提交为止，该操作都将占用操作队列的一项，并以此进行操作数的重命名，保证指令的顺序执行，精确例外处理等。

3) 指令发射

32 位 RISC CPU 采用顺序发射策略，不必等待指令的所有操作数都准备好，指令就可以发射并读取已准备好的寄存器值到相应功能部件的保留站，在各保留站中侦听结果写回总线，等待那些未准备好的操作数。

寄存器堆

32 位 RISC CPU 的定点寄存器堆中包括 32 个 32 位的通用寄存器。

保留站

32 位 RISC CPU 的所有功能部件都分别拥有自己的两项保留站。保留站暂存发射出来的指令，如果指令的所有操作数都准备好，则送往执行部件执行，同时将保留站的这项释放。如果指令的操作数还未准备好，则指令待在保留站中，同时侦听结果写回指令，等待那些未准备好的操作数写回，将其结果直接保存到保留站中，再送往执行部件执行。

4) 指令执行

32 位 RISC CPU 拥有三个执行部件，包括定点 ALU 部件、乘除法部件和访存部件。指令执行阶段各部件执行相应的指令操作，然后将结果写回操作队列中暂存。

定点 ALU 部件

定点 ALU 部件一拍执行完相应操作并且把结果写回，包括加减运算指令、逻辑运算指令、移位指令、转移指令等定点指令。

乘除法部件

乘除法部件实现了全流水的乘法功能和不流水的除法功能。

访存部件

访存部件包括了地址计算、控制寄存器、TLB 和数据 CACHE 三部分。一条访存指令首先经过地址计算获得访存的虚地址，然后通过 TLB 将虚地址转换成物理地址，最后在数据 CACHE 中写入或读出。

数据 CACHE

32 位 RISC CPU 的数据 CACHE 一共 8KB 大小，两路组相联，每路 4KB 大小，每个 CACHE 行大小为 32B。当发生 CACHE 不命中时，从内存中读回对应的 CACHE 行，然后随机替换两路中的一路，将读回的 CACHE 行填入数据 CACHE 中。如果不命中的指令是存数指令，则该指令会暂存在 Buffer 中，允许其它访存指令继续执行，以此实现非阻塞的 CACHE 操作。

5) 指令提交

指令提交就是将操作队列中已经执行完的指令的运算结果写入到定点寄存器堆中，同时将相应的操作队列项释放。如果指令发生异常，则置异常信号，取消处理器中所有指令的执行，并且到相应的异常入口地址取指执行。如果发生转移预测错误，同样取消后续指令的执行，重新开始取指。

4.2 EMI 控制器

EMI 模块为 SOC3210 系列高性能 SOC 芯片的外部存储器访问接口，包括 SDRAM 接口、EEPROM / FLASH 接口、HPI 接口，整个模块通过 HSB 总线与片上 CPU 相连接。外部存储器使用常用的 SDRAM、EEPROM、FLASH 芯片，并支持 VINETIC 芯片。

4.2.1 SDRAM 控制器

SDRAM 控制器并行外接两个 256Mbit SDRAM（位宽 16），这样构成 32bit 数据总线和 EMI 模块相连接。

（1）支持 SDRAM 的命令及其编码

SDRAM 命令由以下信号编码而成（表 4-1)。

CS_ 片选信号

RAS_ 行地址选通信号

CAS_ 列地址选通信号

WE_ 写使能信号

DQM 数据宽度屏蔽信号

ADDR 地址

DQs 数据输出

表 4-1 SDRAM 命令及其编码

命令	CS_	RAS_	CAS_	WE_	DQM	ADDR	DQs
命令禁止（空操作）	X	X	X	X	X	X	
空操作	H	H	H	X	X	X	
激活（选择体，激活行）	L	L	H	H	X	Bank/Row	X
（选择体和列，开始突发式读）	L	H	L	H	L/H	Bank/Col	X
写（选择体和列，开始突发式写）	L	H	L	L	L/H	Bank/Col	Valid
终止突发操作	L	H	H	L	X	X	Active
关闭行（关闭体中的行）	L	L	H	L	X	Code	X
自动刷新或自刷新	L	L	L	H	X	X	X
装载模式寄存器	L	L	L	L	X	Code	X
写允许/输出允许	-	-	-	-	L	-	Active
写禁止/输出高阻	-	-	-	-	H	-	High-Z

（2）发起 SDRAM 初始化操作

在对 SDRAM 进行读写操作之前，首先得对 SDRAM 发起初始化序列，在 SOC3210 系列中，初始化序列是在 SDRAM 配置寄存器控制下，自动对 SDRAM 发起并完成，具体的操作方法如下所述：

首先对 SDRAM 计时寄存器(SD_TIMING)中的具体参数进行配置（各个参数的含义详细信息请参考 SDRAM 标准文档），然后对 SDRAM 模式与大小寄存器（SD_MODE_SIZE）中的具体参数进行配置，同时，把本寄存器中的 PAA 位置 1，在配置完本参数，SDRAM 控制器将自动发起初始化操作。

在对 SDRAM 进行初始化配置之后，用户可以对 SDRAM 空间(0x00000000-0x0fffffff)进行自由访问。

（3）支持的 SDRAM 颗粒类型

SDRAM 控制支持的颗粒类型如下表如列，如果所用颗粒参数不在下表所列范围内，则在配置 SDRAM 控制寄存器时，应采用行列地址数与之相对应的一组参数。

如：所采用颗粒行地址数 12 列地址数 9，颗粒位宽 32,则无法在下表中找到对应类型，此时软件配置控制器时应采用行地址数 12,列地址数 9，颗粒位宽 8 的配置。

表 4-2 SDRAM 控制器支持颗粒类型

	BA 线	行地址数	列地址数	颗粒位宽
颗粒类型 1	2	12	9	8
颗粒类型 2	2	12	10	4
颗粒类型 3	2	12	10	8

	BA 线	行地址数	列地址数	颗粒位宽
颗粒类型 4	2	12	11	8
颗粒类型 5	2	13	9	16
颗粒类型 6	2	13	10	8
颗粒类型 7	2	13	11	4
颗粒类型 8	2	11	9	32
颗粒类型 9	2	10	8	32
其他		11	10	

（4）访问 SDRAM 的时序

本节仅给出 SDRAM 典型的时序关系，

图 4-2、图 4-3 分别为单字读、写时序，图 4-4、图 4-5 分别为多字读、写时序。图中带前缀 H 的信号是 HSB 总线信号，其余为 SOC 引线信号。更详细的引线信号时序图可参阅 SDRAM 芯片的数据手册。

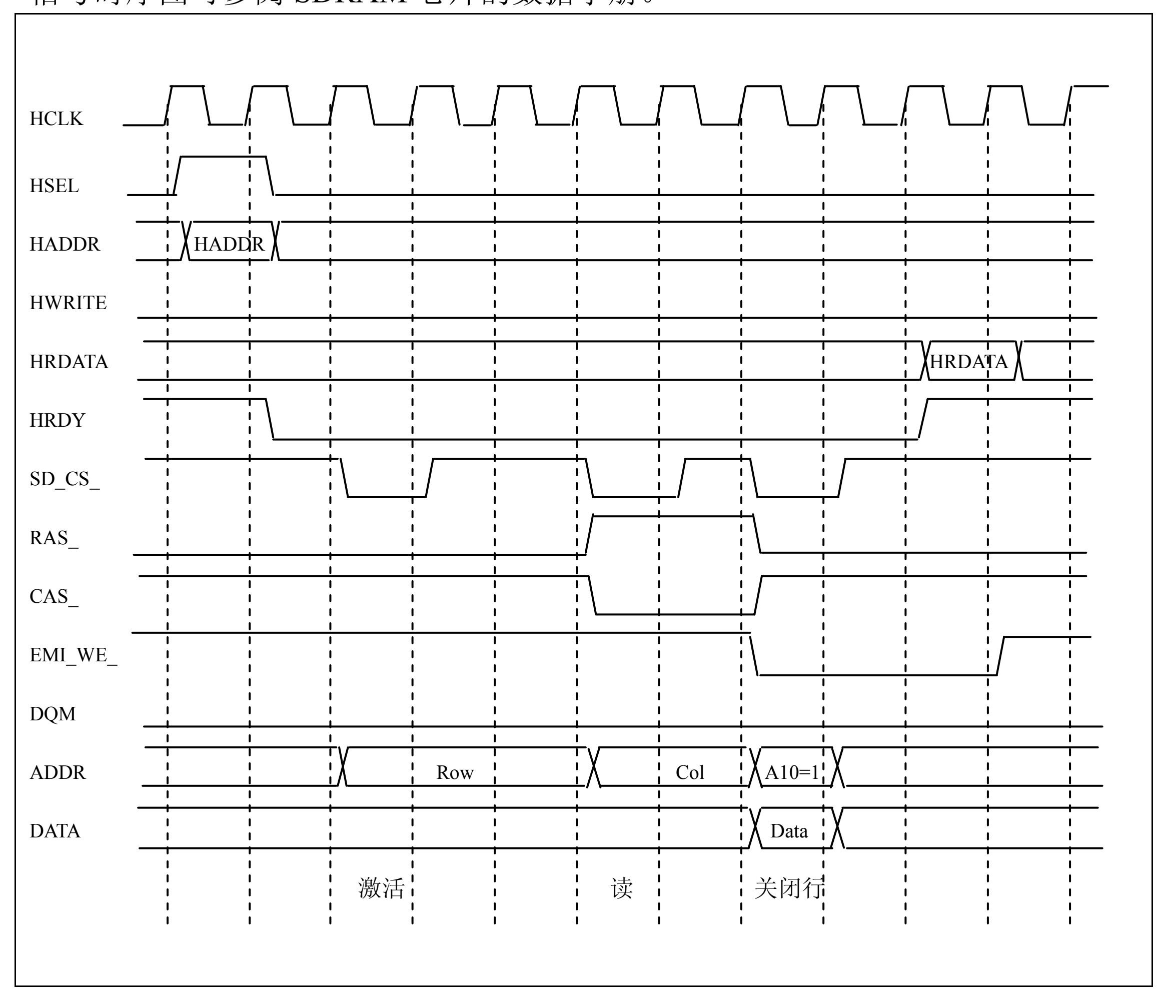


图 4-2 SDRAM 单字读时序图

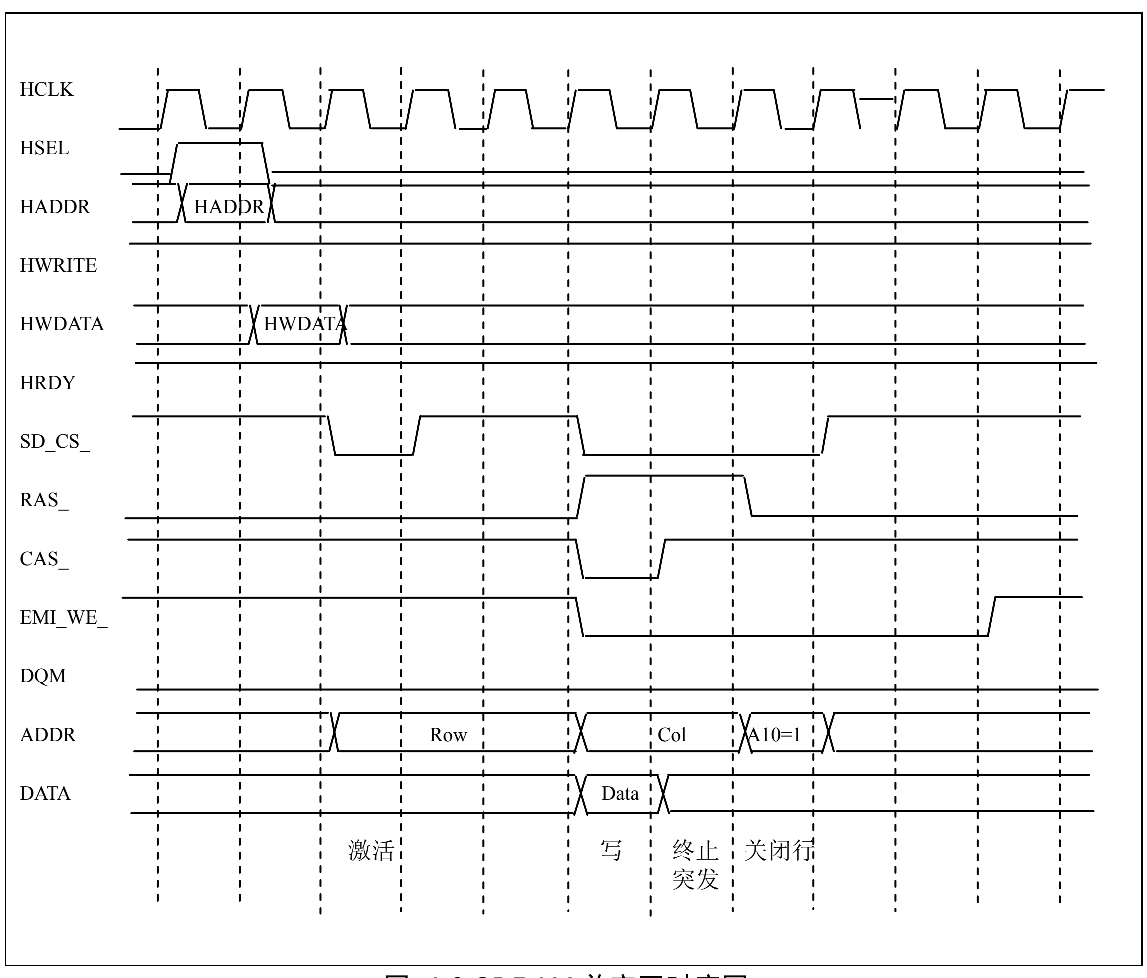


图 4-3 SDRAM 单字写时序图

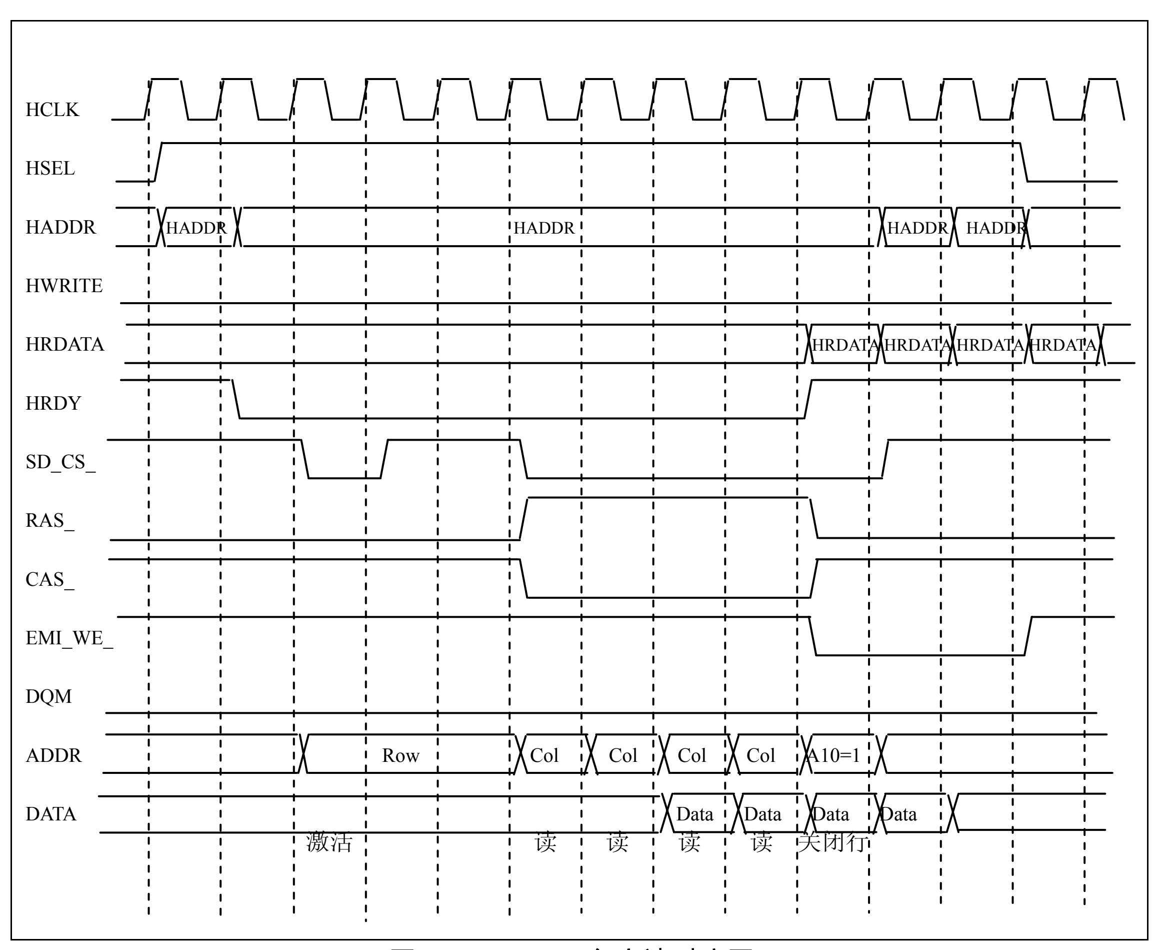


图 4-4 SDRAM 多字读时序图

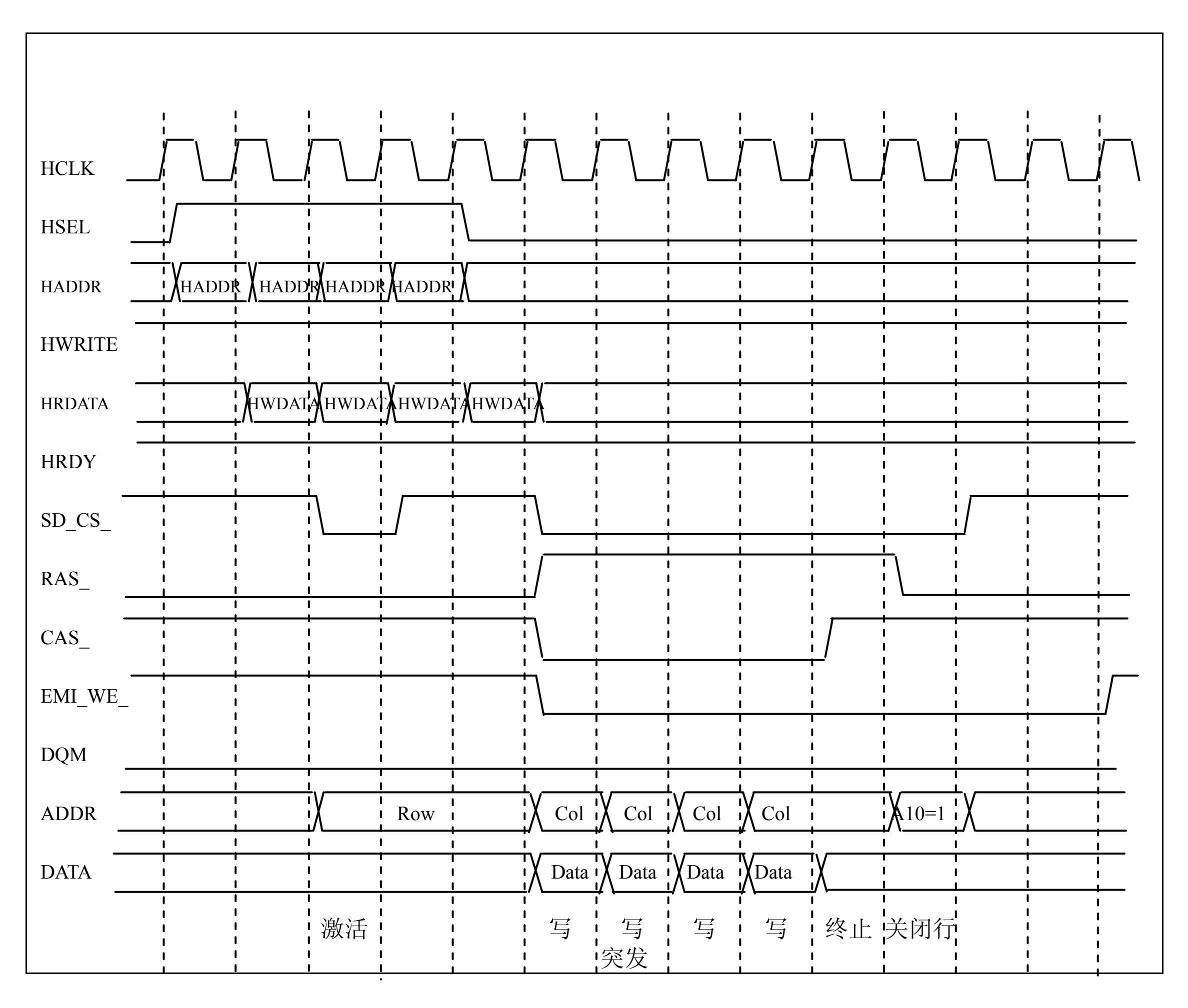


图 4-5 SDRAM 多字写时序图

4.2.2 EEPROM / FLASH 控制器

SOC3210 系列 SOC 支持通过本地 ROM 和 FLASH 引导系统。具体实现时，ROM 采用 EEPROM 来存储引导（Boot）程序，FLASH 可以存储 MCU 执行的固件（Firmware）。下面给出 ROM/FLASH 典型的时序关系（图 4-6~图 4-9），图中带前缀 H 的信号是 HSB 总线信号，其余为 SOC 引线信号。更详细的引线信号时序图可参阅相应的存储器芯片数据手册。

（1）ROM / NOR FLASH 的读写时序

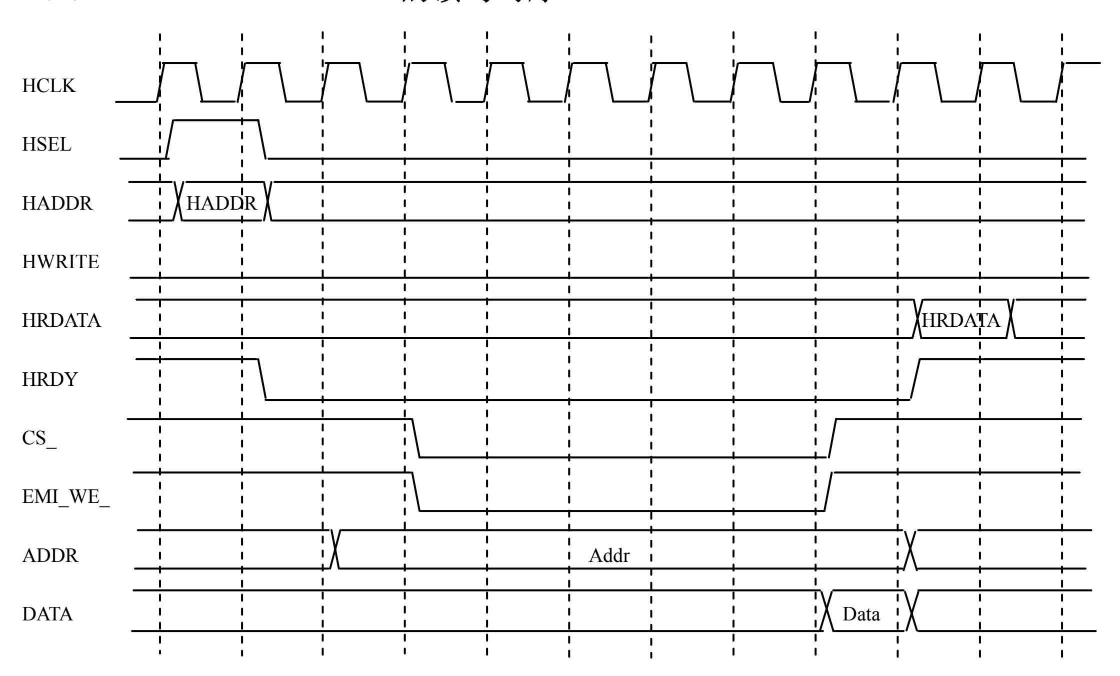


图 4-6 ROM / NOR FLASH 读时序图

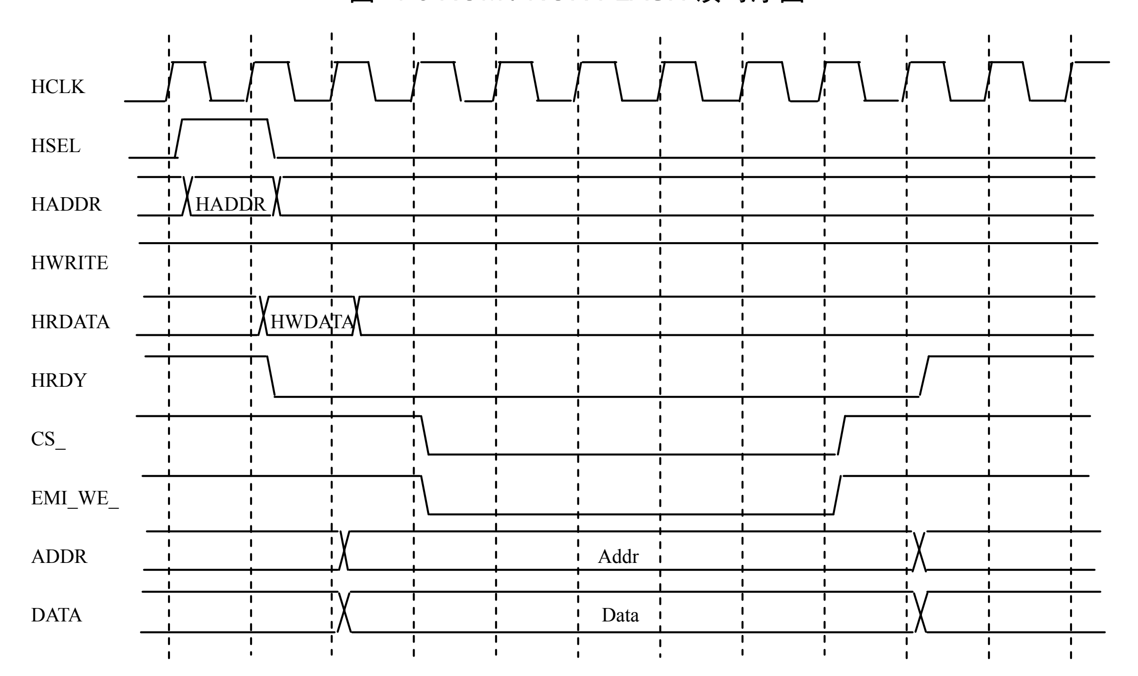


图 4-7 ROM / NOR FLASH 写时序图

（2）NAND FLASH 的读写时序

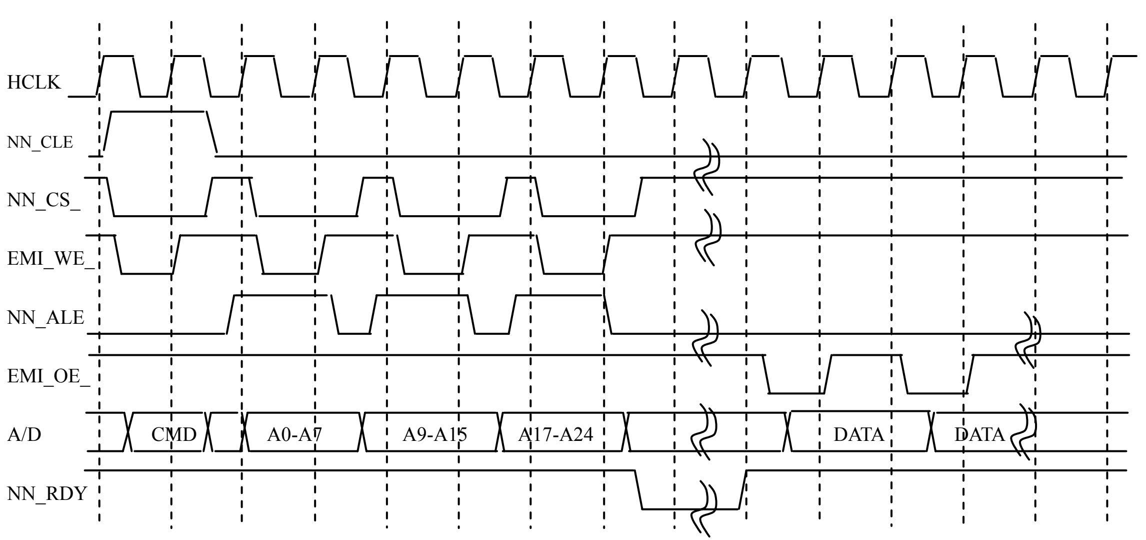


图 4-8 NAND FLASH 整页读时序图

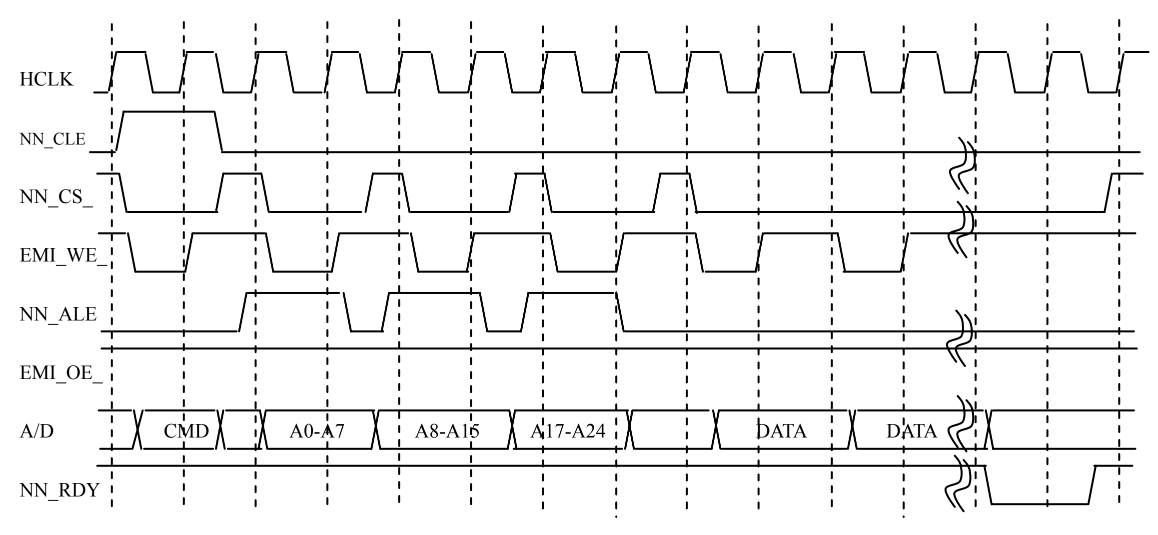


图 4-9 NAND FLASH 整页写时序图

（3）NAND FLASH 的访问控制

NAND FLASH 的读写接口有三个，分别为地址接口（NN_ADDR），命令接口（NN_CMD），数据接口（NN_DATA），对应 NAND FLASH 访问的三种周期，即地址周期，命令周期，数据周期，用户在对 NAND FLASH 进行访问时，需要根据具体使用的 NAND FLASH 对其发出访问序列。每对一个寄存器操作只会产生一次脉冲，也就是说，在 8 位方式下，NN_CMD、NN_ADDR、NN_DATA 只有最低 8 位的数据会被发出，如果是 16 位方式下，NN_CMD、NN_ADDR、NN_DATA 只有最低的 16 位数据/地址会被发出，如果地址周期数为 3/4/5 的话，软件必须分 3/4/5 次分别把每次要发送的地址段写到 NN_ADDR。

例如，对一个图 4-9 中的 NAND FLASH 进行写访问：首先对 NN_CMD 发出相应的写操作，然后分三次对 NN_ADDR 写入相应的地址位，最后，依次向 NN_DATA 发出所需次数的写操作。

上述操作中的宽度为对应 FLASH 的宽度，依据 NN_MODE 中的 NN_WIDTH 配置可以写为 8 位或是 16 位；另外通过配置 NN_HOLD、NN_SPEED 来确定访问 Nand Flash 的时序，其中 NN_HOLD 指的是 NAND 访问时序中，发送完一个地址/数据/命令后需要空的周期数，SPEED 指的是保持控制命令与数据等有效的周期数。

4.2.3 HPI 接口控制器

SOC3210 系列高性能 SOC 可以通过 HPI 接口访问采用相同接口的外围芯片，如 VINETIC 系列芯片，并可以通过不同的访问方式访问。从而达到与外围 DSP 芯片协同工作的目的。其具体接口时序如下图所示：

（1）INTEL DEMULTIPLEXED MODE 的读写时序

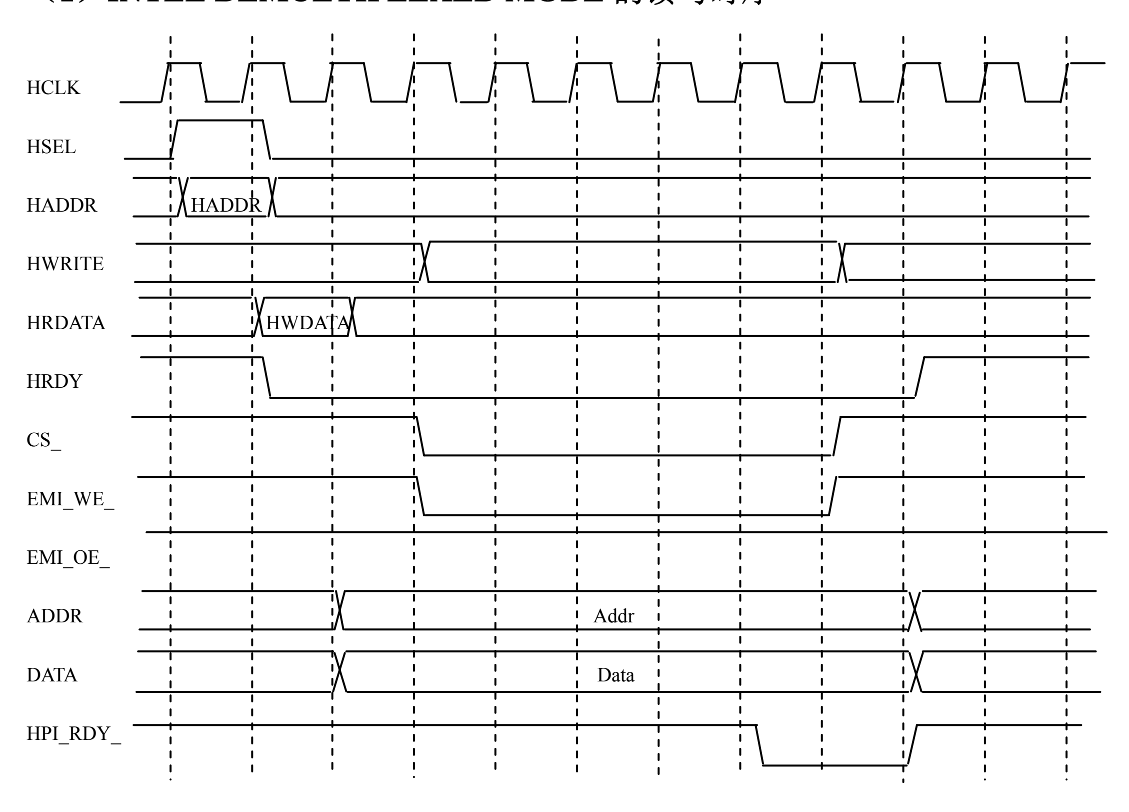


图 4-10 INTEL DEMULTIPLEXED MODE 读写时序图

（2）**MOTOROLA MODE** 的读写时序

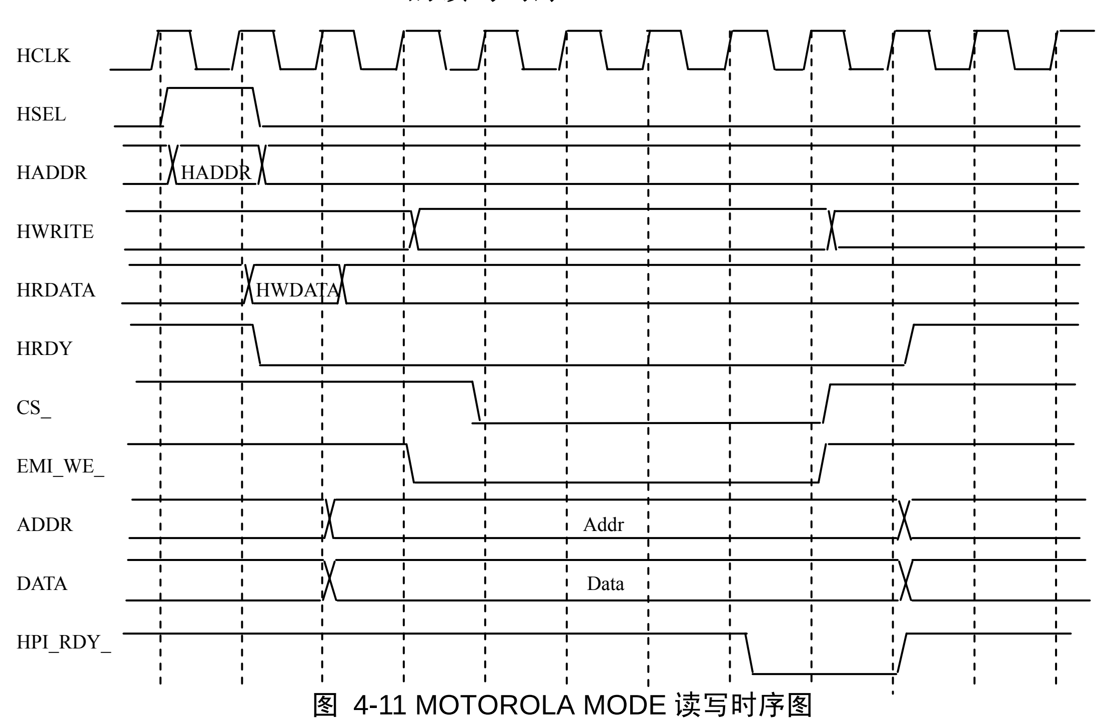


图 4-11 MOTOROLA MODE 读写时序图

4.2.4 EMI 控制器访问地址控制划分

EMI 控制器对外发起访问的地址空间如下表所示：

表 4-3 EMI 地址空间

访问类型	起始地址	结束地址	备注
SDRAM	0x00000000	0x0FFFFFFF	
NAND FLASH 数据	0x1F000040	0x1F000040	数据端口
BOOT ROM	0x1FC00000	0x1FCFFFFF	这部分地址物理上与 NOR FLASH 最低 1M 空间相重合
NOR FLASH	0x18000000	0x19FFFFFF	
HPI	0x1E000000	0x1EFFFFFF	

4.3 LCD 控制器

SOC 集成了一个 LCD 控制器核，核的一端通过内部总线与系统（CPU 核和内存）通信，另一端输出到液晶显示器（含 LCD 驱动器）。

4.3.1 LCD 控制器结构

LCD 控制器核的逻辑结构如图 4-12 LCD 控制器结构所示，各部分的功能说明如下。

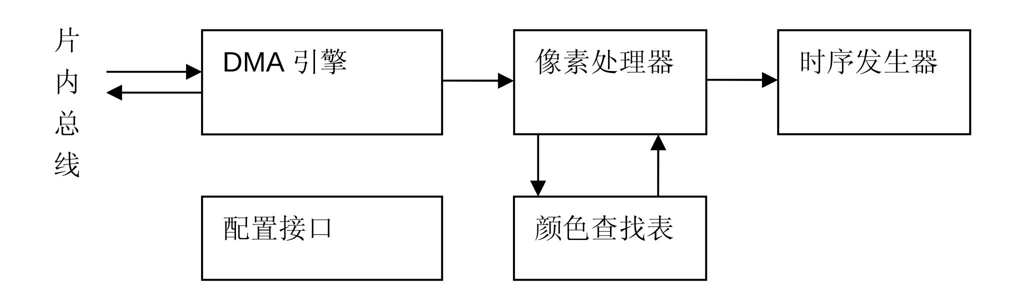


图 4-12 LCD 控制器结构

- **DMA 引擎：**管理所有对外部存储区的访问，读取帧缓冲。
- **像素处理器：**把 32 位像素数据转换为显示屏每个像素点所对应显示信息。色深为 1 位，则把 32 位像素数据从最高位到最低位转换成 32 个像素点信息，每位对应一个像素点信息；色深为 2 位，则把 32 位从最高位到最低位转换成 16 个像素点信息，每 2 位对应一个像素点信息；色深为 4 位，则把 32 位从最高位到最低位转换成 8 个像素点信息，每 4 位对应一个像素点信息；色深为 16 位，则把 32 位从最高位到最低位转换成 2 个像素点信息，每 16 位对应一个像素点信息。在色深为 8 时采用颜色查找表的模式，将 8 位像素点的信息作为颜色查找表的输入，颜色查找表的 18 位输出作为像素点的信息。
- **颜色查找表(Clut)：**颜色查找表用于 8 位色深伪彩色模式。它可以将 8 位的颜色索引转换成 18 位 RGB 颜色数据，实现 256 伪彩色。该查找表共有两个，支持双缓冲机制。
- **时序产生器：**产生驱动 LCD 屏工作的时序信号，包括水平同步脉冲和垂直同步脉冲信号，以及相应的水平中断和垂直中断信号，可视使能信号等。对于不同的工作模式，VGA 模式或者 STN 模式，产生不同的时序信号。

4.3.2 控制器外部接口时序图

（1）VGA 模式

LCD 控制器输出 5 个信号：VCLK，VFRAME，VLINE，VM，VD[7:0]。这些信号协调工作来驱动显示器显示数据，如图 4-13 VGA 模式下的 LCD 视频时序原理图所示。VCLK 是用来驱动显示器的时钟，一直存在，在每个时钟的上升沿，判断可视使能信号 VM 是否有效，如果有效，则根据像素数 VD[7:0] 显示出相

应的点，当遇到行扫描信号 VLINE 时，进行一次行复位工作，当遇到帧扫描信号 VFRAME 时，进行一次整屏刷新。

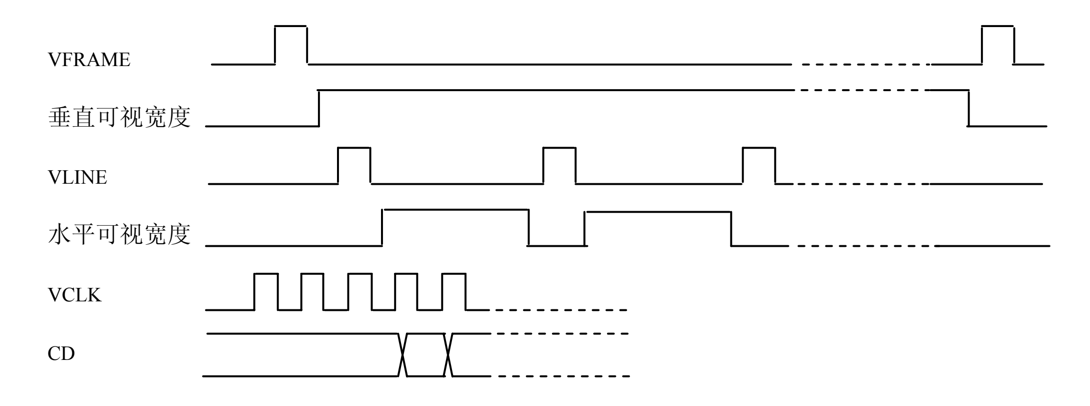


图 4-13 VGA 模式下的 LCD 视频时序原理图

（2）**STN 模式**

当数据准备好时，VCLK 才开始有效，并向 STN LCD 的缓冲区传送数据。VFRAME 脉冲在第一行第一个 VCLK 的上升沿起出现。一行数据传送完成后，行结束脉冲 VLINE 出现，在 VLINE 脉冲结束下降沿，STN LCD 显示该行数据。然后开始新的一行数据传输，如果这是当前帧的第 2 行，此时 VFRAME 脉冲结束（图 4-14）。STN 模式下，VM 用来改变行列电压的极性，而对显示数据是否输出没有影响。

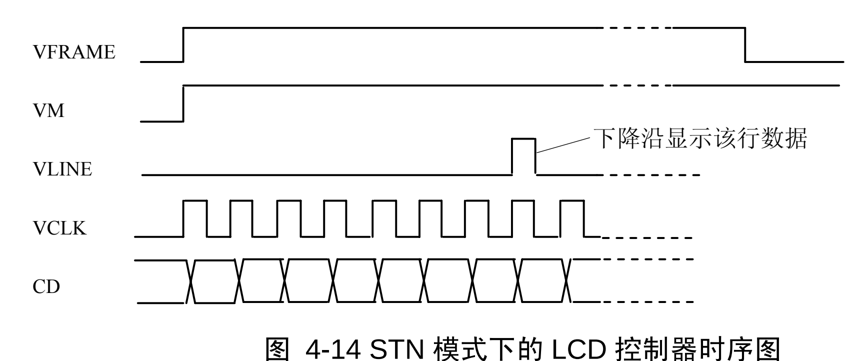


图 4-14 STN 模式下的 LCD 控制器时序图

4.3.3 LCD 控制

SOC3210 系列中 LCD 控制器只提供简单的显示控制功能。使用系统内存作为显存，把已经准备好的帧缓冲数据转换成 LCD 面板的显示信号。其主要工作环境如图 4-15 LCD 控制器工作概要所示。

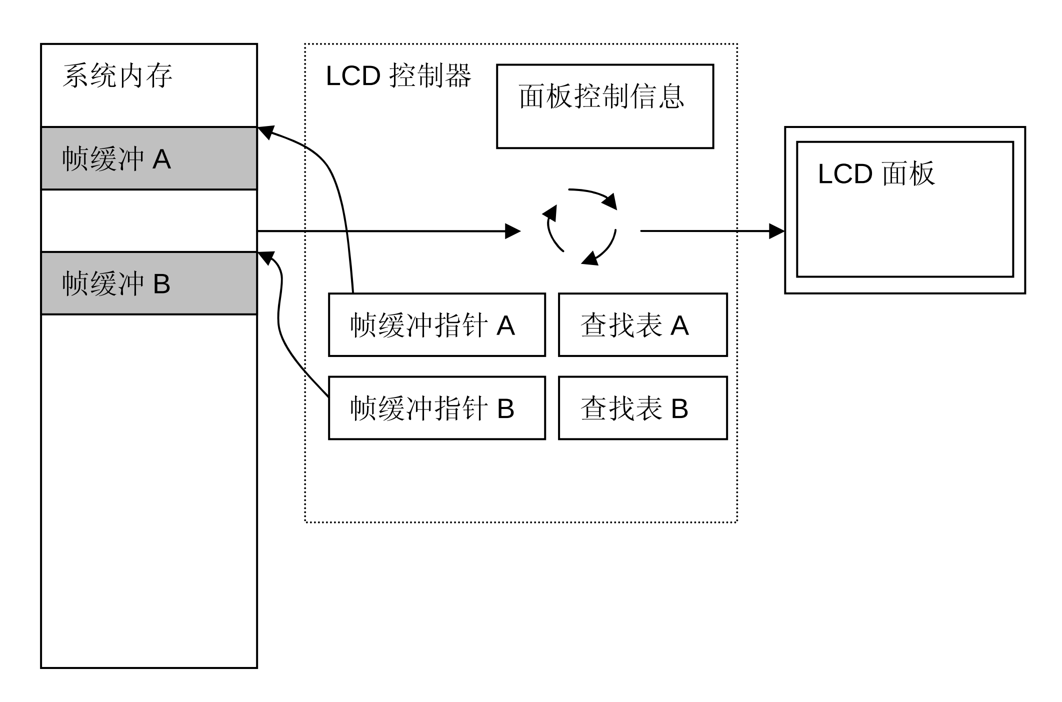


图 4-15 LCD 控制器工作概要

在开始工作之前，软件需要初始化显示缓冲区，并将控制器中的缓冲区首指针设置好。每个 LCD 面板都有特定的扫描时序要求，需要作相应配置。在初始化内部控制寄存器后，LCD 控制器便不断地从帧缓冲取数据，转换成 LCD 的显示格式。取完一屏数据后再从下一个帧的首址开始处理。如果不使用双缓冲等功能，软件在初始化控制器后就不需理会控制器，简单地把像素点信息写在显示缓冲区就可以在屏幕上显示。

（1）启动与停止

LCD 控制器的工作由 LCD_CTRL 的 VEN 控制，除了表切换控制（CBSWE, VBSWE）的更改不需要先停止 LCD 外，其它控制位的改变必须先让 LCD 控制器停下。例如从当前控制字 0x00000781 更改至 0x00000701，需要先写 0x00000700(除最低位为 0 其它都等于目标值)，然后才把最低位置 1 后写到 CTRL 寄存器。

（2）突发读长度

突发读长度（VBL）决定 LCD 控制器一次读所取回的数据量(4*n 字节)。取回的越多，总线传输的效率越高，对片上其他设备的影响越小。由于 LCD 控制器按行取显示数据，一行的数据量必须能被一次取回的量所整除。即：显示 64 像素一行，色深 16 位 时，一行数据量为 64*2=128 字节。如果用 64 拍的传

输，一次会读回 256 字节数据，不能整除 128，故不可用。合适的值应该是 n=8(VBL=10b)。

（3）帧缓冲及查找表切换

利用双缓冲可以实现显示图像的完整显示。当前显示帧缓冲由 AVMP 指示，正在绘制的图像在另一个帧缓冲。当准备完成后发送 VBSWE 切换信号到 LCD 控制器，则显示帧切换。若允许帧切换中断，控制器会在真正开始读新的帧数据时发 VBSINT 中断。查找表的切换与帧切换类似，为使显示图像完整，硬件保证查找表切换只会在开始处理新一帧时进行。

（4）帧缓冲及查找表中数据表示

LCD 控制器用小尾端解释显示缓冲区中的数据，访问的粒度与一个像素所占的字节数相同。TFT 模式下显示输出为 18 位的 VD 信号，从高往低对应 R、G、B 3 个颜色分量。

假设当前显示的帧缓冲区首指针为 0，对应数据如下：

地址	0	1	2	3
数据	a	b	c	d

显示输出见表 4-4。

表 4-4 LCD 显示输出

显示模式	输出像素	输出数据 vd[17:0]
1bpp	a[7] a[6] a[5] … b[7] b[6] … d[0]	1 位扩展成相同的 18 位 {18{a[7]}}
2bpp	a[7:6] a[5:4] a[3:2] a[1:0] ... d[3:2] d[1:0]	每个通道输出相同的值 像素值高位置于输出通道的高位，低位复制 5 份填入输出的低位 {3{a[7], 5{a[6]}}}
4bpp	a[7:4] a[3:0] b[7:4] … d[3:0]	送到输出的高 4 位，低 2 位写 0 {3{a[7:4], 2’b00}}
8bpp pseudo	a[7:0] b[7:0] c[7:0] d[7:0]	取自查找表 clut[acmp][a[7:0]]
16bpp	{b[7:0],a[7:0]} {d[7:0],c[7:0]}	6 位绿色分量不变，红、蓝分量低位补 0 后输出 {{b[7:3], 1’b0}, // R {b[2:0], a[7:5]},// G {a[4:0], 1’b0}} // B

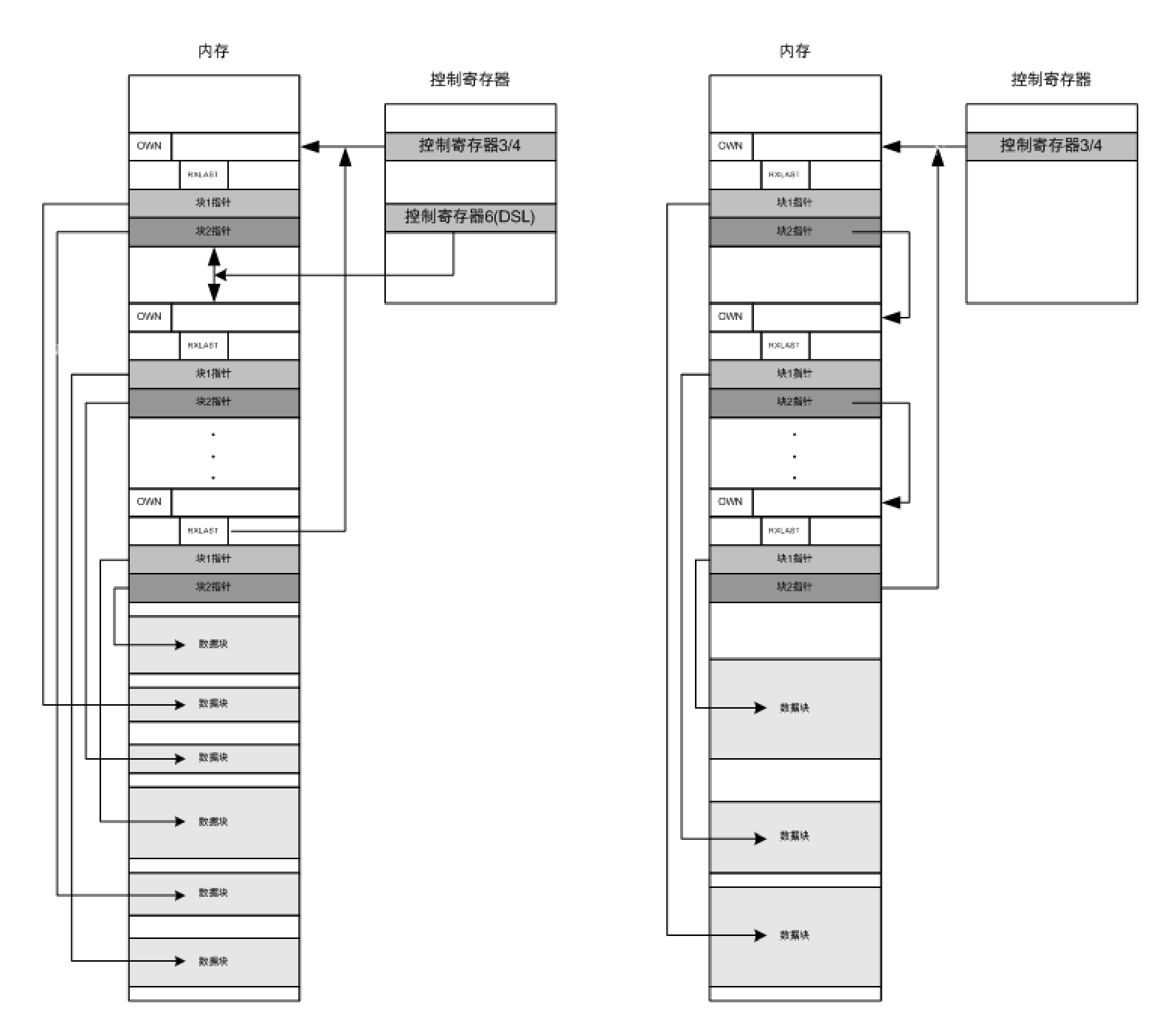
STN 模式下帧缓冲数据的解释与 TFT 模式相同，只是输出的值按 STN 的

约定，从高往低循环排列各个像素的数据后输出。

4.4 以太网支持

系统的以太网部分包含两个MAC模块，分别为MAC0和MAC1。两个模块的功能完全一致，均以HSB接口接入到系统的HSB总线上。MAC模块是一个完整的10/100M以太网MAC层控制器。能够完成IEEE 802.3协议中对10/100M以太网的媒体访问控制层所定义的载波监听多路访问/冲突检测（CSMA/CD）算法。其通过一系列控制和状态寄存器同CPU进行交互，在CPU的控制下完成帧的接收和发送。MAC模块同SDRAM之间的数据传输是通过DMA控制器完成的。在传输数据时，DMA控制器以HSB总线的主设备身份访问SDRAM，从发送数据缓冲区取出数据或是将数据写入接收数据缓冲区，这个过程不需要CPU的介入，从而得到了高的传输效率。

下面将对MAC模块的基本操作流程做进一步介绍。CPU与MAC模块的交互有两套接口，其中一套是通过对MAC模块中的控制寄存器进行操作，同时从MAC模块的状态寄存器中获取MAC的相关状态，从而做出下一步操控。关于寄存器的具体介绍，见本文档的第三部分MAC控制器寄存器，其中与上行口相关的见3.12.1～3.12.11，与下行口相关的见3.12.13～3.12.24。另一套重要的接口是在双方共享的存储空间中建立的数据结构——描述符链表和数据缓存，用于存放帧的数据和与帧相关的信息。数据缓存中存放着CPU准备发送的数据或是从MAC接收来的数据。描述符链表是一个指针链表，其中的指针指向各个数据缓存块。发送和接收各自维护一条描述符链表，由CPU负责在系统初始化的时候进行创建，链表的长度任意。其中接收描述符链表第一项所在内存地址保存在MAC的控制状态寄存器3中，发送描述符链表第一项所在内存地址保存在MAC的控制状态寄存器4中。描述符链表有两种组织形式可供选择，分别是“链”模式和“环”模式。在“链”模式下，每个描述符中都存有一个指针，指向链表中下一个描述符的地址。在“环”模式下，链表中下一个描述符同当前描述符末尾的地址间隔是固定的，存放在0号控制状态寄存器中的DSL域中。每个描述符最多可以包含两个数据缓存指针，但是当采用“链”模式时候，其中一个用作指向下一个描述符，故只有一个指针可用于指向数据缓存。一帧可以占据多个描述符和多个数据缓冲块，但是一个描述符最多只能对应一帧。具体结构见图 4-16。



描述符链表“环”模式　　　　描述符链表“链”模式

图 4-16 MAC 描述符链表结构

每个描述符中都包含一个 OWN 位，用来标示该描述符以及所指向的数据缓冲块是属于 CPU 还是 MAC。在正式的接收、发送过程开始之前，CPU 必须创建好接收和发送描述符链。初始时，所有接收描述符的 OWN 都应该设成属于 MAC，所有的发送描述符都应该设成属于 CPU。当 CPU 通知 MAC 开始发送，MAC 取回发送描述符链中当前待处理的那一个描述符，检查其 OWN 位。如果该描述符已经被设置成属于 MAC，那么该描述符所指向的数据缓冲块的内容被 MAC 的 dma 读出，然后发送出去。如果这一帧占用了多个描述符，那么接下来的一个描述符被读回，重复上述过程。描述符中的 FS（First Descriptor）位为 1 表示该描述符是一帧的第一个描述符，LS（Last Descriptor）位为 1 表示该描述符是一帧的最后一个描述符。当 CPU 通知 MAC 可以开始接收帧时，MAC 取回接收描述符链中当前待处理的那一个描述符，检查它的 OWN 位是否设成属于 MAC。如果是，并且接收到的一帧通过了 MAC 地址过滤，那么帧的内容被写入到该描述符指向的数据缓存中。如果一个描述符放不下这一帧，那么该帧将被依次放在若干个描述符中。其中第一个描述符的 FS 位置 1，最后一个描

述符 LS 位置 1。整个描述符的结构见图 4-17。

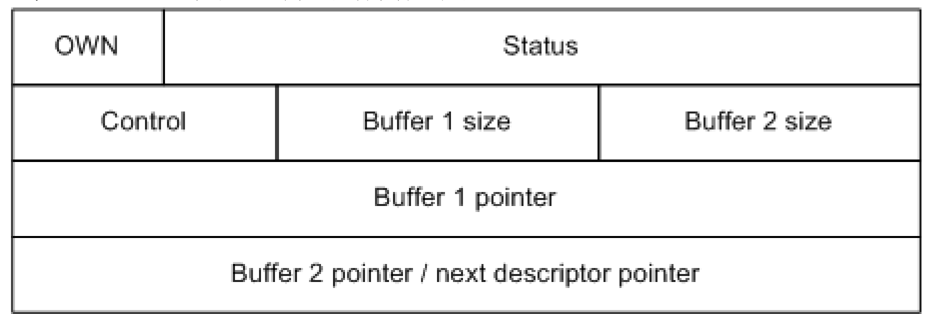


图 4-17 MAC 描述符结构

下面对接收描述符（类似图 4-17 结构）的每个域做描述：

表 4-5 接收描述符第 0 项

位	符号	描述
31	OWN	1—MAC 占有该描述符 0—CPU 占有该描述符 如果当前帧已经接收完毕，或是该描述符指向的数据缓冲资源已经用完，MAC 都会将这一位清 0
30	AFF	地址过滤失败。 当置为 1 的时候，表明接收的这一帧没有通过 MAC 地址识别过程。 该位有意义当且仅当该描述符第 0 项 LS 位，MAC 的 CSR6.30（receive all）位都置为 1，且帧的长度至少有 64 字节。
29..16	FRL	帧长度。单位：字节。 该域的值有意义仅在该描述符第 0 项的 LS 位为 1，DE 位为 0 时
15	ERR	出错总标志 当描述符第 0 项的 CRCE、COL、FTL、RUNT、DESE 位中有一位置起时，该位为 1。表示接收的帧有错误。 该位有意义当且仅当该描述符第 0 项的 LS 位为 1
14	DESE	描述符错误 当 MAC 在接收帧的过程中，没有可用的接收数据缓冲资源可用的时候，MAC 会将该位置为 1。 该位有意义当且仅当该描述符第 0 项的 LS 位为 1
11	RUNT	残帧标志 为 1 时表示在冲突窗口（64 字节）结束之前，接收帧的过程就因为网络上的冲突而提前结束。 该位有意义当且仅当该描述符第 0 项的 LS 位为 1
10	MCF	多播帧标志 为 1 时表示这一帧的目的地址是一个组播地址。 该位有意义当且仅当该描述符第 0 项的 LS 位为 1
9	FS	首描述符标志 为 1 时表示该描述符是一帧的第一个描述符
8	LS	尾描述符标志 为 1 时表示该描述符是一帧的最后一个描述符

位	符号	描述
7	FTL	帧超长标志 为 1 时表示这一帧的长度超过 1518 字节。 该位有意义当且仅当该描述符第 0 项的 LS 位为 1
6	COL	冲突标志 为 1 时表示在接收超过接收窗口（64 字节）后检测到冲突。 该位有意义当且仅当该描述符第 0 项的 LS 位为 1
5	TYPE	帧类型 为 1 表示这一帧是以太网格式的帧，为 0 表示这一帧是 802.3 格式的帧。 该位有意义当且仅当该描述符第 0 项的 LS 位为 1
3	MIIE	MII 接口出错标志 为 1 时表示 MII 接口上发现错误。 该位有意义当且仅当该描述符第 0 项的 LS 位为 1
2	DRB	dribbling bit 为 1 时表示帧不是字节对齐的。 该位有意义当且仅当该描述符第 0 项的 LS 位为 1
1	CRCE	CRC 校验出错标志 为 1 时表示接收帧的 CRC 校验出错。 该位有意义当且仅当该描述符第 0 项的 LS 位为 1。并且当接收帧是一个残帧的时候，该位也没有意义
0	OV	接收 FIFO 溢出标志 为 1 时表示在接收过程中，接收 FIFO 溢出

表 4-6 接收描述符第 1 项

位	符号	描述
25	RXLAST	接收描述符环的最后一个 为 1 时表示该描述符是接收描述符环的最后一个。当 MAC 对此描述符操作完毕后，自动的跳回到描述符环的第一个。
24	RLINK	为 1 时表示描述符的第 3 项存放的是下一个描述符的地址，而不是数据缓存的地址
21..11	RBS2	该描述符指向的第二块数据缓冲块的大小，以字节为单位。这里的数值必须是 4 的倍数。如果这一项为 0，则 MAC 忽略第二块数据缓冲块，直接取下一个描述符。 该域有意义仅当描述符第 1 项的 RCH 位为 0
10..0	RBS1	该描述符指向的第一块数据缓冲块的大小，以字节为单位。这里的数值必须是 4 的倍数。如果这一项为 0，则 MAC 忽略第一块数据缓冲块，使用第二块数据缓冲块。

表 4-7 接收描述符第 2 项

位	符号	描述
31..0	RBA1	第一块数据缓冲块的地址 该地址必须字对齐，即最低两位必须为 0

表 4-8 接收描述符第 3 项

位	符号	描述
31..0	RBA2	第二块数据缓冲块的地址 该地址必须字对齐，即最低两位必须为 0

下面是发送描述符（类似图 4-17 结构）：

表 4-9 发送描述符第 0 项

位	符号	描述
31	OWN	1—MAC 占有该描述符 0—CPU 占有该描述符 如果当前帧已经发送完毕，或是该描述符指向的数据缓冲中的内容已经发送完，MAC 都会将这一位清 0
15	ERR	出错总标志位 当描述符第 0 项的 UF、EC、LC、NC、LO 位中有一位为 1，则该位置为 1。 该位有意义仅当描述符第 1 项的 LS 位为 1
11	LM	为 1 时表示在发送过程中失去了传输介质的使用权。 该位有意义仅当描述符第 1 项的 LS 位为 1
10	NM	为 1 时表示在发送过程中，没有传输介质。 该位有意义仅当描述符第 1 项的 LS 位为 1
9	LCOL	为 1 时表示在发送出 64 字节后检测到冲突。 当描述符第 0 项的 UF 位为 1，该位无意义； 该位有意义仅当描述符第 1 项的 LS 位为 1
8	ECOL	为 1 时表示发送在连续 16 次重传尝试后放弃。 该位有意义仅当描述符第 1 项的 LS 位为 1
6..3	COLC	冲突次数记录 记录下一帧发送完毕之前所遇到的冲突次数。 当描述符第 0 项的 EC 位为 1，该位无意义； 该位有意义仅当描述符第 1 项的 LS 位为 1
1	UNF	为 1 时表示在发送一帧的过程中，发送 FIFO 变空。 该位有意义仅当描述符第 1 项的 LS 位为 1
0	DEFER	为 1 表明该帧在发送之前退避过。 该位有意义仅当描述符第 1 项的 LS 位为 1

表 4-10 发送描述符第 1 项

位	符号	描述
31	TINT	帧发送结束中断 该位置 1，通知 MAC 当前一帧发送完毕后，立刻将 CSR5 中的发送中断位置起。 该位有意义仅当描述符第 1 项的 LS 位为 1，或是发送设置帧
30	LS	为 1 表示是一帧的最后一个描述符
29	FS	为 1 表示是一帧的第一个描述符
28	FIL1	地址过滤类型标识 这一位同 FIL0 位一起，决定了当前帧的地址过滤类型。 仅对于设置帧有意义

位	符号	描述
27	SETF	为 1 表示这一帧是一个设置帧
26	ACRC	为 1 时，MAC 不为一帧自动添加 CRC。 唯一的例外是，当发送帧的内容小于 64 字节，且自动加 PAD 的功能开启，MAC 将在填充 PAD 后自动的加上 CRC，不管该位是否设置。
25	TXLAST	为 1 时表示这是发送描述符环的最后一个描述符
24	TLINK	为 1 时表示该发送描述符的第 3 项存放的是下一个描述符的地址，而不是数据缓冲块的地址。
23	DPAD	为 1 时，MAC 关闭为小于 64 字节的帧自动添加 PAD 的功能。 当 MAC 为一帧添加 PAD 时，无论 AC 位设置为何，最后的 CRC 都会添加上
22	FIL0	地址过滤类型标识 这一位同 FT1 位一起，决定了当前帧的地址过滤类型。 仅对于设置帧有意义
21..11	TBS2	该描述符的第二块数据缓冲块中使用的空间大小，以字节为单位。如果这一项为 0，则 MAC 忽略第二块数据缓冲块，直接取下一个描述符。 本位有意义仅当描述符第 1 项 TCH 位为 0 时
10..0	TBS1	该描述符的第一块数据缓冲块中使用的空间大小，以字节为单位。如果这一项为 0，则 MAC 忽略第二块数据缓冲块，直接访问第二块数据缓冲块。

表 4-11 发送描述符第 2 项

位	符号	描述
31..0	TBA1	第一块数据缓冲块的地址。对于设置帧，这一地址必须字对齐，即最低两位为 0。在其它情况下，没有特殊要求。

表 4-12 发送描述符第 3 项

位	符号	描述
31..0	TBA1	第二块数据缓冲块的地址。没有特殊要求。

为了完成 MAC 控制器的 MAC 地址过滤功能，需要对 MAC 进行相应的配置。这里 MAC 模块不是通过直接配置寄存器进行的，而是通过“发送”一种特殊的设置帧来完成，这一帧的内容并不真正发送到以太网络上，仅仅是写入到 MAC 的地址过滤 RAM 中。一个有效的设置帧必须 192 字节长，必须存放在单独一个数据缓冲块中，且字对齐。同时，发送描述符中的 FS 位和 LS 位都要置 0。设置帧的发送描述符中的 FIL1 和 FIL0 决定了过滤的模式，见表 4-13。表 4-14 列出了完整过滤模式下设置帧的格式，表 4-15 列出了哈希过滤模式下设置帧的格式。

表 4-13 地址过滤模式

FIL1	FIL0	描述
0	0	完整过滤模式 设置帧被解释成 16 个 48 位的 MAC 地址

FIL1	FIL0	描述
0	1	混合哈希过滤模式 设置帧被解释成包含 512 位的 hash 表，并添加一个 48 位的 MAC 地址
1	0	倒置完整过滤模式 设置帧被解释成 16 个 48 位的 MAC 地址
1	1	哈希过滤模式 设置帧被解释成包含 512 位的 hash 表

表 4-14 完整过滤模式设置帧格式

byte	data bits 31:16	data bits 15:0
3:0	xxxxxxxxxxxxxxxx	MAC 地址 0(15 : 0)
7:4	xxxxxxxxxxxxxxxx	MAC 地址 0(31:16)
11:8	xxxxxxxxxxxxxxxx	MAC 地址 0(47:32)
12:9	xxxxxxxxxxxxxxxx	MAC 地址 1(15 : 0)
16:13	xxxxxxxxxxxxxxxx	MAC 地址 1(31:16)
20:17	xxxxxxxxxxxxxxxx	MAC 地址 1(47:32)
• • •	• • •	• • •
171:168	xxxxxxxxxxxxxxxx	MAC 地址 14(15 : 0)
175:172	xxxxxxxxxxxxxxxx	MAC 地址 14(31:16)
179:176	xxxxxxxxxxxxxxxx	MAC 地址 14(47:32)
183:180	xxxxxxxxxxxxxxxx	MAC 地址 15(15 : 0)
187:184	xxxxxxxxxxxxxxxx	MAC 地址 15(31:16)
191:188	xxxxxxxxxxxxxxxx	MAC 地址 15(47:32)

表 4-15 哈希过滤模式设置帧格式

byte	data bits 31:16	data bits 15:0
3:0	xxxxxxxxxxxxxxxx	Hash filter(15 : 0)
7:4	xxxxxxxxxxxxxxxx	Hash filter(31:16)
11:8	xxxxxxxxxxxxxxxx	Hash filter(47:32)
• • •	• • •	• • •
123:121	xxxxxxxxxxxxxxxx	Hash filter(495:480)
127:124	xxxxxxxxxxxxxxxx	Hash filter(511:496)
• • •	• • •	• • •
159:156	xxxxxxxxxxxxxxxx	MAC 地址 (15 : 0)
163:160	xxxxxxxxxxxxxxxx	MAC 地址 (31:16)
167:164	xxxxxxxxxxxxxxxx	MAC 地址 (47:32)
171:168	xxxxxxxxxxxxxxxx	xxxxxxxxxxxxxxxx
175:172	xxxxxxxxxxxxxxxx	xxxxxxxxxxxxxxxx

byte	data bits 31:16	data bits 15:0
• • •	• • •	• • •
183:180	xxxxxxxxxxxxxxxx	xxxxxxxxxxxxxxxx
187:184	xxxxxxxxxxxxxxxx	xxxxxxxxxxxxxxxx
191:188	xxxxxxxxxxxxxxxx	xxxxxxxxxxxxxxxx

下面介绍一些基本的处理流程：

- 初始化流程

  1、创建接收描述符链表，并为其分配数据缓冲块资源；（记住所有的 own 位置 1）

  2、创建发送描述符链表，并为其分配数据缓冲块资源；（记住所有的 own 位置 0）

  3、配置 CSR

  a) 写接收描述符链表基址寄存器（CSR3）；

  b) 写发送描述符链表基址寄存器（CSR4）；

  c) 配置 CSR0 中的 DMABL、PRI、TXAP，确定数据传输接口；

  d) 写 CSR6，配置双工模式（FD），发送限度（TRC, TTM, SF）以及接收地址过滤方式（RALL, PR, PMC, PBF）

  e) 写 CSR7 和 CSR11，配置中断处理

  4、创建并“发送”设置帧；

  5、使能发送操作；

  6、使能接收操作；

- 接收处理流程

  1、检查当前描述符的 own 位；

  2、如果描述符属于 CPU：

  a) 将帧的数据送交上层处理；

  b) 将描述符设置成属于 MAC；

- 发送处理流程

  1、检查当前描述符的 own 位；

  2、如果描述符属于 MAC，则报告没有空闲的描述符；

  3、否则，设置数据缓冲块的长度；

  4、设置数据缓冲块的地址；

  5、将描述符的 own 位设成属于 MAC；

- 中断处理流程

  1、检查中断请求源

a) 如果是接收中断，调用接收处理流程；

b) 如果是发送中断，释放发送帧的数据缓冲块；

c) 如果是异常中断，调用错误处理程序；

2、清中段请求源

➢ 模式监控流程

1、通过 MII 接口的 SMI 读取 PHY 的状态；

2、如果没有 link 上：

a) 报告没有连接

b) 将 CSR6.TXS 和 CSR6.RXS 清 0

c) 转到 5

3、如果 PHY 处于 FD 模式下，但 MAC 处于 HD 模式

a) 报告：设置 FD 模式

b) 等待 MAC 处于停止状态（CSR5.TXST=000 且 CSR5.RXST=000）

c) 设置 FD 选择位为 1（CSR6.FD）

4、如果 PHY 处于 HD 模式下，但 MAC 处于 FD 模式

a) 报告：设置 HD 模式

b) 等待 MAC 处于停止状态（CSR5.TXST=000 且 CSR5.RXST=000）

c) 设置 FD 选择位为 0（CSR6.FD）

5、挂起本线程，等待操作系统触发下一个线程；

在上面的“模式监控流程”中，需要能知晓 PHY 的状态，对 PHY 的控制就只能依靠软件通过 SMI 接口完成。SMI 接口包含两个信号，一个为控制时钟 MDC，另一个为数据输入/输出 MDIO。因在芯片内部无三态逻辑存在，故实际软件可见的是 MDC、MDI、MDO 和 MDEN 四个信号。对其操作可通过寄存器的 MAC0_SMI_EEPROM_CTL 和 MAC1_SMI_EEPROM_CTL 的 19 位至 16 位完成。

对 PHY 进行操作的数据以特殊的帧格式进行传输。每一帧都包含前导、操作码、PHY 地址、寄存器偏移和传送的数据。前导是 32 个连续的 1，之后紧接着两位的 01 表示一帧的开始。PHY 地址能够指定本次传输针对同一 MII 接口上的哪个 PHY，若 MAC 只连接了一个 PHY，所以这个 PHY 地址是固定的。寄存器偏移能指定这次访问 PHY 中 32 个 16 位寄存器的哪一个。至于对于这些寄存器的访问是写还是读，由操作码决定，10 表示读操作，01 表示写操作。图 4-18 和图 4-19 分别给出了 SMI 接口上进行读操作和写操作的时序。在读操作过程中，当操作码和地址都给出后，存在 2 位的翻转时隙，用于防止 MDIO 上传输变向产生的信号冲突。具体 PHY 中寄存器的含义请参考选用外置的 PHY

芯片的 datasheet 了。

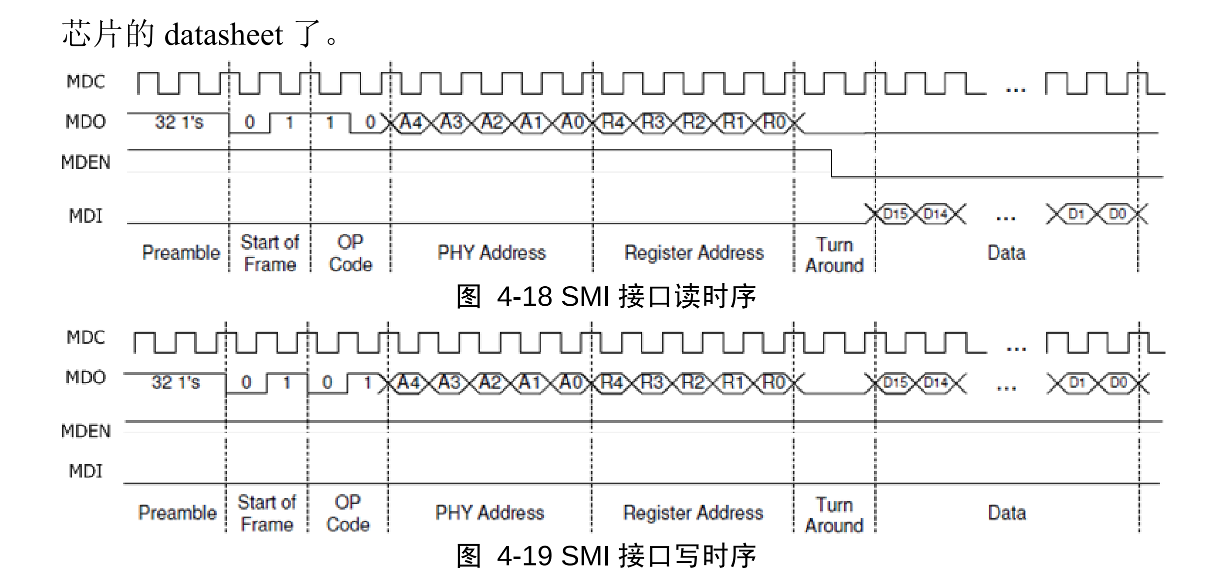


图 4-18 SMI 接口读时序

图 4-19 SMI 接口写时序

4.5 AC97 控制器

在系统里一个 AC97 应用系统如图 4-20 所示。在一个片上系统中，与 AC97 控制器相连的有 3 部分：一是外设总线，接收来自微处理器的控制信息以及配置信息；二是 AC97 Codec，多媒体数字信号编解码器，该解码器对 PCM 信号进行调制，输出人耳接受的模拟声音或者把真实的声音转换为 PCM 信号，转换通过 D/A 转换器以及 D/A 转换器实现；三是 DMA 引擎，通过 DMA 的方式写或读 AC97 控制器内部的 FIFO，实现 PCM 音频数据的不间断操作。DMA 是通过微处理器配置的，从微处理器设定的内存区域搬运数据给 FIFO 或者把 FIFO 的数据搬运到设定的内存区域。

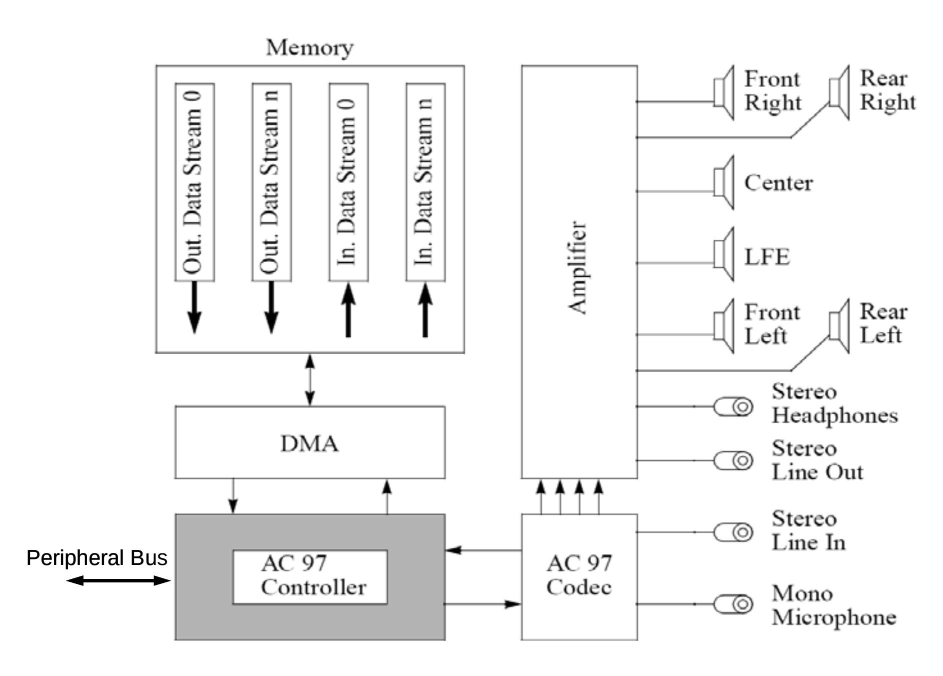


图 4-20 AC97 应用系统

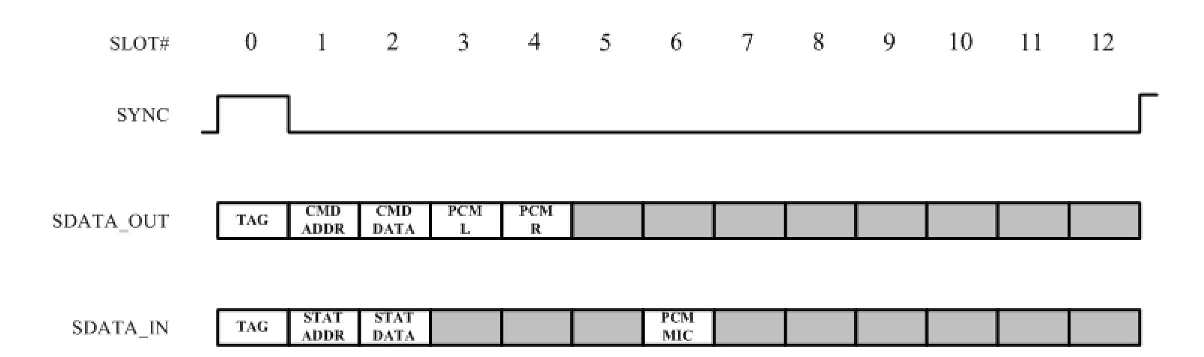
4.5.1 AC97 控制器特性

AC97 控制器支持 2 个输出通道和 1 个输入通道，支持控制一个 AC97 Codec。它完全符合 AC97 2.2 规范，只支持 AC97 的音频解码。以下是它的主要特性：支持可变的采样率，最高达到 48KHz 的速率。

1. 支持 8，16 位采样宽度。
2. 两个输出端口，支持双声道输出。
3. 支持单声道 MIC 输入。
4. 支持双倍采样率。
5. 支持外部 DMA 引擎。

4.5.2 AC-Link

AC-link 是连接 AC97 Codec 和控制器的数字串行接口（5 线）。AC-link 协议是双向、固定时钟频率的串行数字信号。AC-link 通过时分复用（TDM）将音频帧分成 12 个输出和 12 个输入数据 slot，每个 slot 有 20 位数据，以此来处理多个输入和输出 PCM 音频流以及控制寄存器的访问。SOC3210 系列的 AC97 控制器的 BIT_CLK 来自于外部的 CODEC。AC_Link 所用到的 Slot 如图 4-21 所示：



AC_Link Output Slots：

Slot	Name	Description
0	SDATA_OUT TAG	MSBs 表示哪些 slot 有效; LSBs 表示 Codec ID
1	Control CMD ADDR	Read/write 命令加 7-bit Codec 寄存器地址
2	Control DATA	16-bit 命令寄存器写地址
3,4	PCM L&R DAC	16, 18, or 20-bit PCM 左右声道数据

AC_Link Input Slots：

Slot	Name	Description
0	SDATA_IN TAG	MSBs 表示哪些slot有效
1	STATUS ADDR	MSBs为寄存器地址s; LSBs表示哪些slot需要数据
2	STATUS DATA	16-bit命令寄存器读数据

6	Dedicated Microphone	16, 18 or 20-bit PCM数据

图 4-21 AC_Link 帧的 Slot 分配

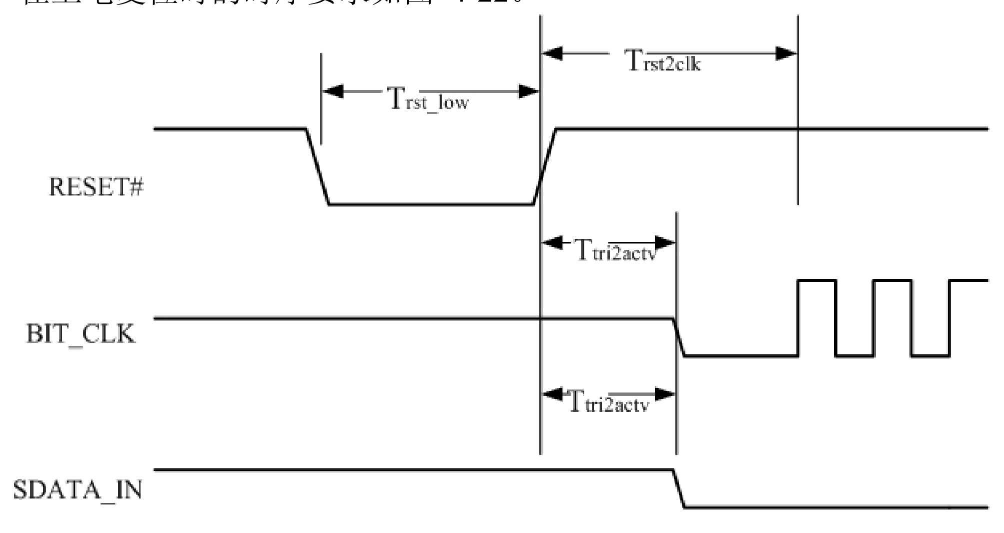
4.5.3 Reset

AC97 规范定义了三类 Reset 过程，其中 Cold Reset 和 Warm Reset 的时序分别见图 4-22 和图 4-23。

类型	途径	描述
Cold reset	RESET#	完全的硬件复位，如上电复位
Register reset - Audio	write to 00h	向 CSR 寄存器的 0 位写入复位使能，所有 audio 寄存器复位
Warm reset	SYNC	重起 AC-link，寄存器值不改变

- **Cold Reset**

在上电复位时的时序要求如图 4-22。

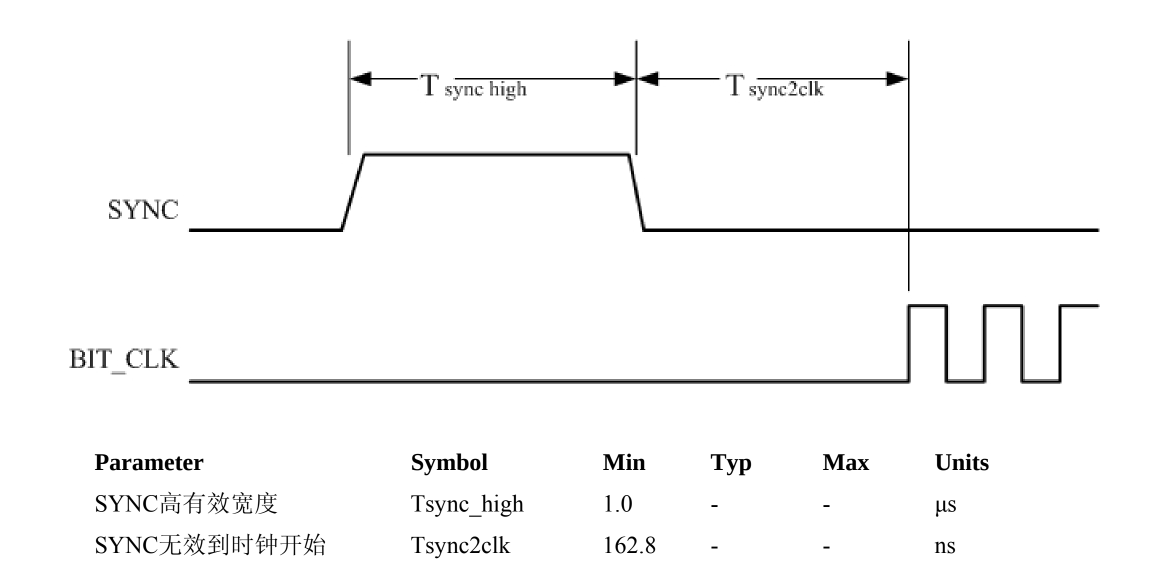


Parameter	Symbol	Min	Typ	Max	Units
RESET#有效宽度	Trst_low	1.0	-	-	µs
RESET#无效到三态撤销	Ttri2actv	-	-	25	ns
RESET#无效到时钟开始	Trst2clk	162.8	-	400	ns

图 4-22 Cold Reset 时序图（BIT_CLK 由 CODEC 提供）

- **Warm Reset**

Warm 复位时序如图 4-23 所示。



Parameter	Symbol	Min	Typ	Max	Units
SYNC高有效宽度	Tsync_high	1.0	-	-	µs
SYNC无效到时钟开始	Tsync2clk	162.8	-	-	ns

图 4-23Warm Reset 时序图

4.5.4 AC97 控制器工作过程

在图 4-20 中可以看到，AC97 控制器是通过软件控制 DMA 控制器、AC97 控制器、CODEC 来实现音频的输入和输出，具体过程如下：

初始化

AC97 的驱动先对 DMA、AC97 控制器和 CODEC 进行初始化。

1）配置输出通道配置寄存器 OCC0

2）配置输入通道配置寄存器 ICC

3）通过 Codec 访问寄存器 CRAC 来对 Codec 芯片内部寄存器进行配置。CPU 读 Codec 寄存器时，要等 Codec 通过输入帧返回相应的数据，然后中断通知 CPU 读回。CPU 写 Codec 寄存器时，当要写的数据已经发送到输出帧以后，也会中断通知 CPU。在 CPU 请求了对 Codec 寄存器读写后，在收到中断以前，不允许再进行新的读写操作。

工作过程

AC97 音频系统的工作是通过 CPU、DMA、AC97 控制器协同工作来完成，具体过程如下简介：

1）配置初始化

CPU 先配置 3.11 节中 DMA 的相关寄存器，OUT_MODE 寄存器和中断屏蔽寄存器 INT_MASK。

2）音频输出

CPU 计算完一块音频数据，就向 AC97 DMA 控制器的 OUT_ADDR 和 OUT_LENGTH 分别写入数据起始地址和数据长度，DMA 控制器里有深度为 2 的 FIFO 来存储这两组数据，DMA 接受到起始地址和长度后就开始启

动 DMA 功能，当 AC97 控制器每帧向 DMA 请求数据时，DMA 就启动 HSB 请求去 memory 空间读数据，同时长度递减，当长度减到 0 后，DMA 向 CPU 发送中断，CPU 向 DMA 送新的地址和长度，如此反复。

3）音频输入

对于 MIC 的输入过程，也如 2）类似，当程序启动音频输入后，CPU 就将地址和长度写入 IN_ADDR 和 IN_LENGTH，DMA 启动，当 AC97 控制器每帧接受完新的输入数据后就通知 DMA 控制器，DMA 控制器去 AC97 控制器的 FIFO 里读取数据，通过 HSB 送到 memory 空间，同时长度递减，当一段数据接受完成后就向 CPU 发送中断，CPU 向 DMA 送新的地址和长度，如此反复。

4.6 SPI 控制器

串行外围设备接口 SPI 总线技术是 Motorola 公司推出的多种微处理器、微控制器以及外围设备之间的一种全双工、同步、串行数据接口标准。SPI 总线是一种三线总线，因其硬件功能很强。

4.6.1 SPI 主控制器结构

本系统集成的仅为主控制器，所连接的是从设备，主要用于客户显示屏。图 4-24 是 SPI 主控制器的结构，系统寄存器包括控制寄存器，状态寄存器和外部寄存器，分频器生成 SPI 总线工作的时钟信号，由于数据读、写缓冲器（FIFO）允许 SPI 同时进行串行发送和接收数据。

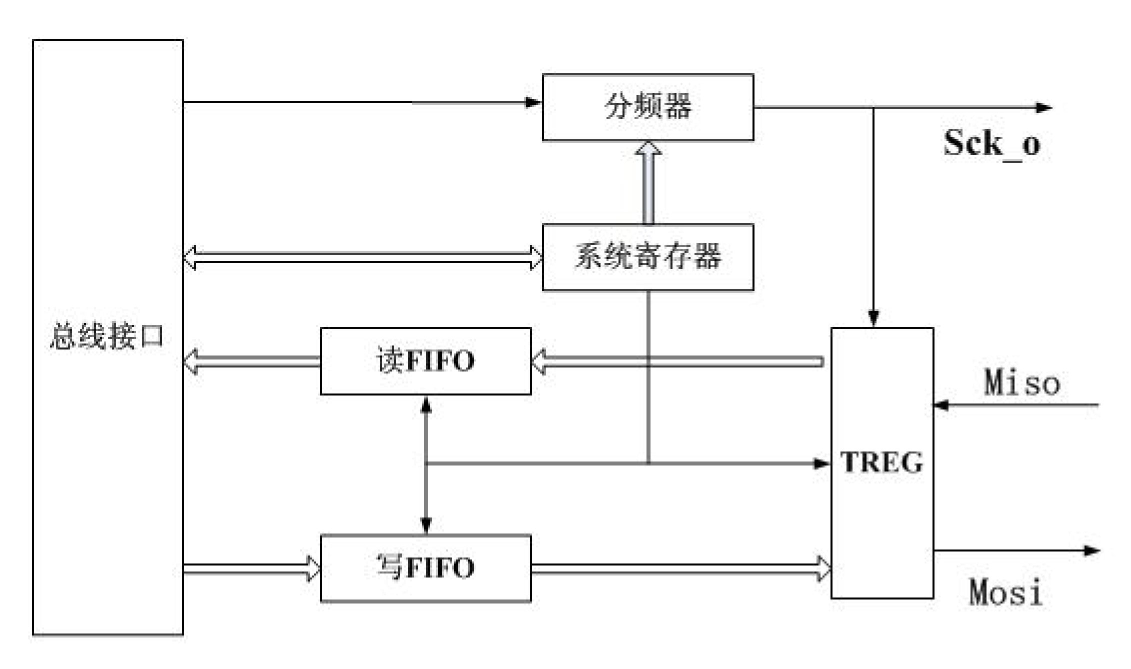


图 4-24 SPI 主控制器结构

4.6.2 SPI 主控制器外部接口时序图

如图 4-25 所示，SPI 主控制器发送数据时，数据提前半拍放在 MOSI 引线上，接着从设备端用时钟边沿锁存数据。根据时钟极性（CPOL）和时钟相位（CPHA）的设定，有 4 种可能的时序关系。

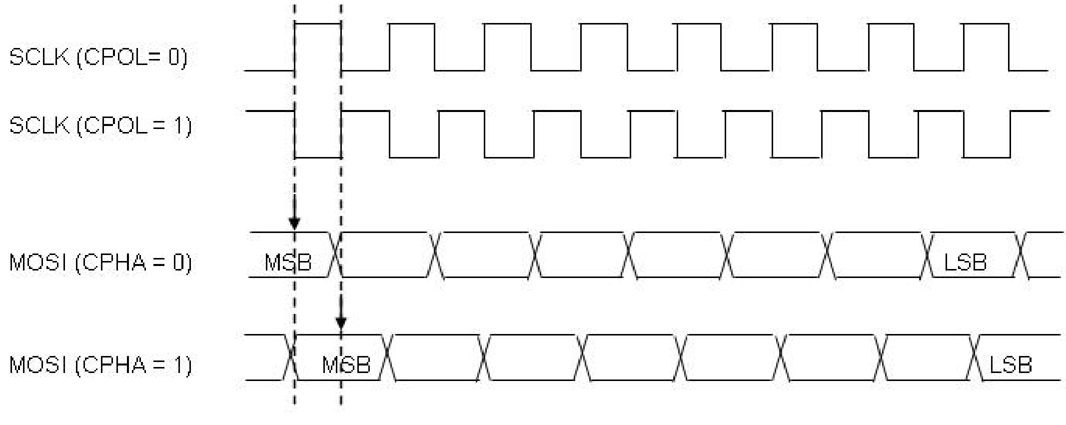


图 4-25 SPI 主控制器时序图

4.6.3 SPI 主控制器的读写操作

1. 模块初始化

- 复位外部寄存器sper，对寄存器写入8'b0000_0000
- 重置状态寄存器spsr，对寄存器写入8'b1100_0000

2. 模块的发送/传输操作

- 停止 SPI 控制器工作，对控制寄存器 spcr 写入 8'b1001_0100
- 重置状态寄存器 spsr，对寄存器写入 8'b1100_0000
- 设置外部寄存器 sper，包括中断申请条件 sper[7:6]和分频系数 sper[1:0]，具体参考寄存器说明
- 启动 SPI 控制器，对控制寄存器 spcr 写入 8'b1101_0100
- 往数据传输寄存器写入数据

3. 中断处理

- 接收到中断申请
- 读状态寄存器 spsr 的值，若 spsr[2]为 1 则表示数据发送完成，若 spsr[0]为 1 则表示已经接收数据
- 写寄存器数据或者读寄存器数据
- 往状态寄存器 spsr 的 spif 位写 1，清除控制器的中断申请

4.7 PS/2 控制器

PS/2接口用于许多键盘和鼠标设备，最初是由IBM公司开发的。PS/2接口的物理连接器有4条引线，地线、+5 V、数据、时钟。

4.7.1 PS/2 控制器结构

PS/2 控制器的结构如图 4-26 所示。

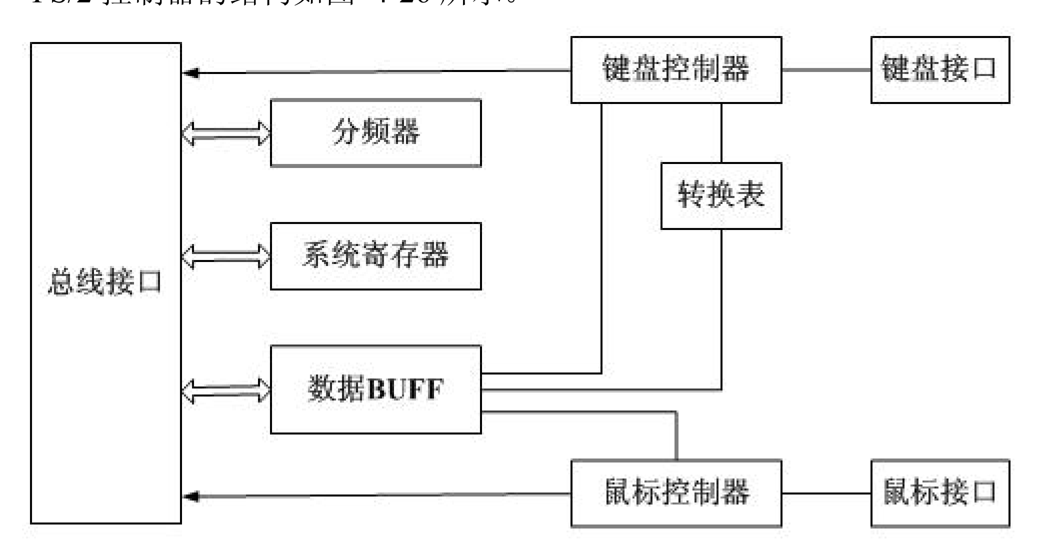


图 4-26 PS/2 控制器结构

PS/2 键盘和鼠标履行一种双向同步串行协议，换句话说每次数据线上发送一位数据，并且每次在时钟线上发一个脉冲，数据就被读入。键盘/鼠标可以发送数据到主机，而主机也可以发送数据到设备，但主机总是在总线上有优先权，它可以在任何时候抑制来自于键盘/鼠标的通讯，只要把时钟拉低即可。

所有传送的数据为字节数据，每个字节构成一帧，包含11或12 位。表 4-16 是字节帧定义。

表 4-16 SPI 数据帧

1 start bit	起始位，恒为0
8 data bits	8个数据位，低位在前
1 parity bit	校验位，奇校验
1 stop bit	停止位，恒为1
1 acknowledge bit	应答位，仅用于主机到设备的通讯

4.7.2 PS/2 控制器外部接口时序图

（1）设备到主机的通讯过程：

数据和时钟线都是集电极开路结构（正常保持高电平）。当键盘或鼠标等待发送数据时，首先检查时钟线的电平。如果是低电平，则是主机抑制了通讯设备，必须缓冲任何要发送的数据直到重新获得总线的控制权（键盘有16 字节的缓冲区，而鼠标的缓冲区仅存储最后一个要发送的数据包）。如果时钟线是高电平，设备就可以开始传送数据。仅当设备时钟脉冲为高时改变数据线，而在时钟脉冲下降沿时数据被锁存（图 4-27）。

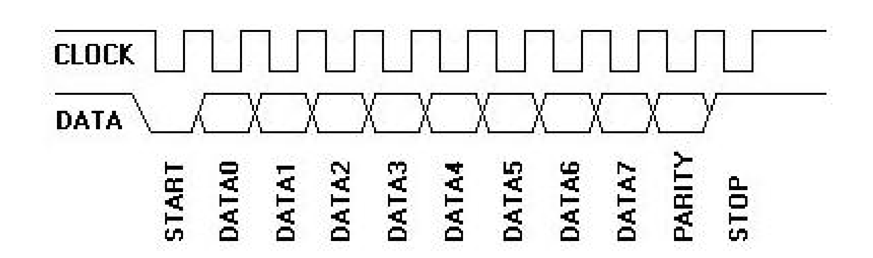


图 4-27 PS/2 设备到主机通讯的时序图

（2）主机到设备的通讯过程：

这时数据包发送稍有不同。首先PS/2设备总是产生时钟信号，如果主机要发送数据，它必须首先把时钟和数据线设置为“请求发送”状态。即通过下拉时钟线抑制通讯，通过下拉数据线实施“请求发送”，然后释放时钟线。图 4-28(a) 是主机和设备分解的时序图，图 4-28 (b) 是合成的时序图。

当设备检测到这个“请求发送”状态，它将开始产生时钟信号，用时钟脉冲输入8个数据位和1个停止位。主机仅当时钟脉冲为低时改变数据线，而在时钟脉冲上升沿时数据被锁存。

在停止位发送后，设备要应答接收到的字节，就把数据线拉低，并产生最后一个时钟脉冲。如果主机在第11个时钟脉冲后不释放数据线，设备将继续产生时钟脉冲直到数据线被释放（然后设备将产生一个错误）。

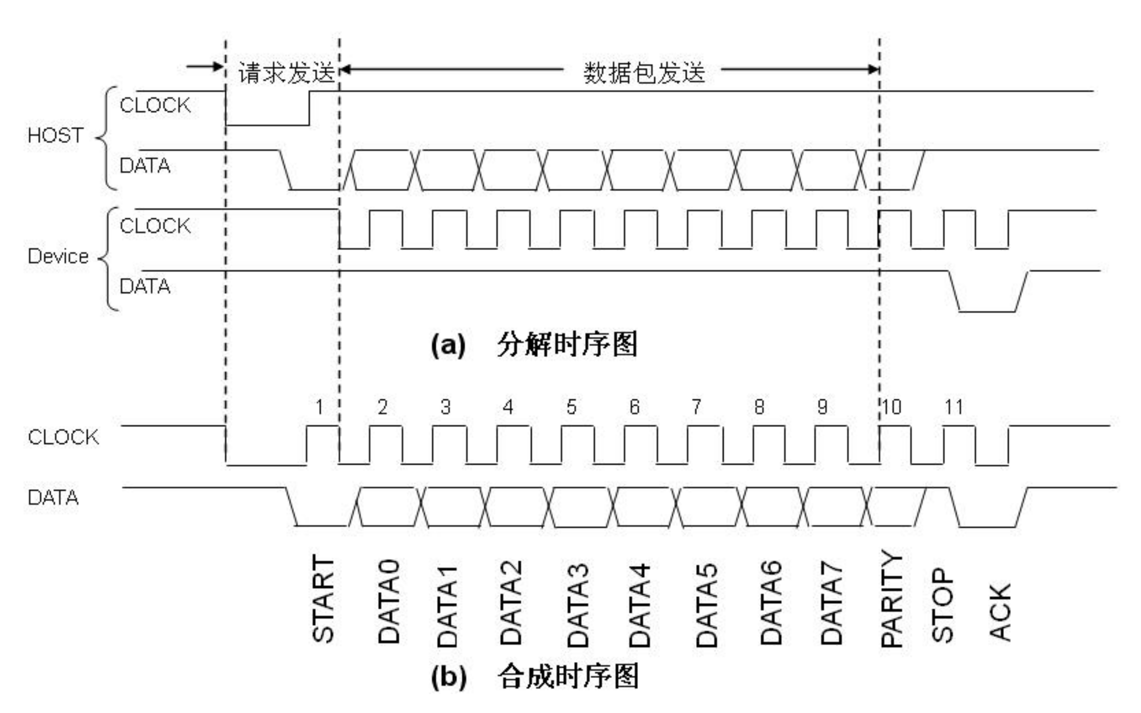


图 4-28 PS/2 主机到设备通讯的时序图

4.7.3 PS/2 主控制器的读写操作

1. 模块初始化

- 设置分频寄存器，包括 5usec 分频寄存器和 60usec 分频寄存器

- 给命令寄存器键盘送入控制器自检命令，值为 8’hAA
- 读取检测结果，值是否为 8’h55，若是则正确
- 给命令寄存器键盘送入控制器自检命令，值为 8’hAB
- 读取检测结果，值是否为 8’h00，若是则正确
- 给命令寄存器键盘送入键盘使能，值为 8’hAE
- 往数据传输寄存器写入数据使键盘复位，值为 8’hFF
- 等待键盘中断
- 读取状态寄存器 RSR 并判断该寄存器的 IBF 位，为 1 表示键盘数据输入
- 读取接收数据寄存器，并判断值是否为 8’hFA，自动清除中断
- 重复前面 4 个步骤，依次往键盘写入 8’hED，8’h07，8’hED，8’h00，
- 若有鼠标，则设置鼠标参数包括采用频率，分辨率、缩放比例和使能数据报告

2．模块的读写操作

A．读数据

- 等待中断申请并判断中断源是 K_INT 还是 M_INT
- 读取状态寄存器 RSR 并判断该寄存器的 IBF 位和 AIBF 位，IBF 为 1 表示键盘数据输入，AIBF 为 1 表示鼠标数据输入
- 读取接收数据寄存器，控制器自动清除中断申请

B．写数据

若接到了键盘的三个特殊按键（CapsLock，NumLock 和 ScrollLock）时需要往键盘回写数据。

- 读取状态寄存器 RSR 并判断 OBF 位，为 0 表示当前发送数据寄存器没有数据
- 往数据传输寄存器写入数据。

4.8 UART 控制器

系统SOC集成了一个UART核，通过LPB总线与总线桥通信。UART控制器提供与MODEM或其他外部设备串行通信的功能，例如与另外一台计算机，以RS232为标准使用串行线路进行通信。 该控制器在设计上能很好地兼容国际工业标准半导体设备16550A。

4.8.1 UART 控制器结构

UART 控制器有发送和接收模块（Transmitter and Receiver）、MODEM 模块、中断仲裁模块（Interrupt Arbitrator）、和访问寄存器模块（Register Access Control），这些模块之间的关系见图 4-29 所示。主要模块功能及特征描述如下：

- **发送和接收模块：**负责处理数据帧的发送和接收。发送模块是将 FIFO 发送队列中的数据按照设定的格式把并行数据转换为串行数据帧，并通过发送端口送出去。接收模块则监视接收端信号，一旦出现有效开始位，就进行接收，并实现将接收到的异步串行数据帧转换为并行数据，存入 FIFO 接收队列中，同时检查数据帧格式是否有错。UART 的帧结构是通过行控制寄存器（LCR）设置的，发送和接收器的状态被保存在行状态寄存器（LSR）中
- **MODEM 模块：**MODEM 控制寄存器（MCR）控制输出信号 DTR 和 RTS 的状态。MODEM 控制模块监视输入信号 DCD,CTS,DSR 和 RI 的线路状态，并将这些信号的状态记录在 MODEM 状态寄存器（MSR）的相对应位中。
- **中断仲裁模块：**当任何一种中断条件被满足，并且在中断使能寄存器（IER）中相应位置 1，那么 UART 的中断请求信号 UAT_INT 被置为有效状态。为了减少和外部软件的交互，UART 把中断分为四个级别，并且在中断标识寄存器（IIR）中标识这些中断。四个级别的中断按优先级级别由高到低的排列顺序为，接收线路状态中断；接收数据准备好中断；传送拥有寄存器为空中断；MODEM 状态中断。
- **访问寄存器模块：**当 UART 模块被选中时，CPU 可通过读或写操作访问被地址线选中的寄存器。

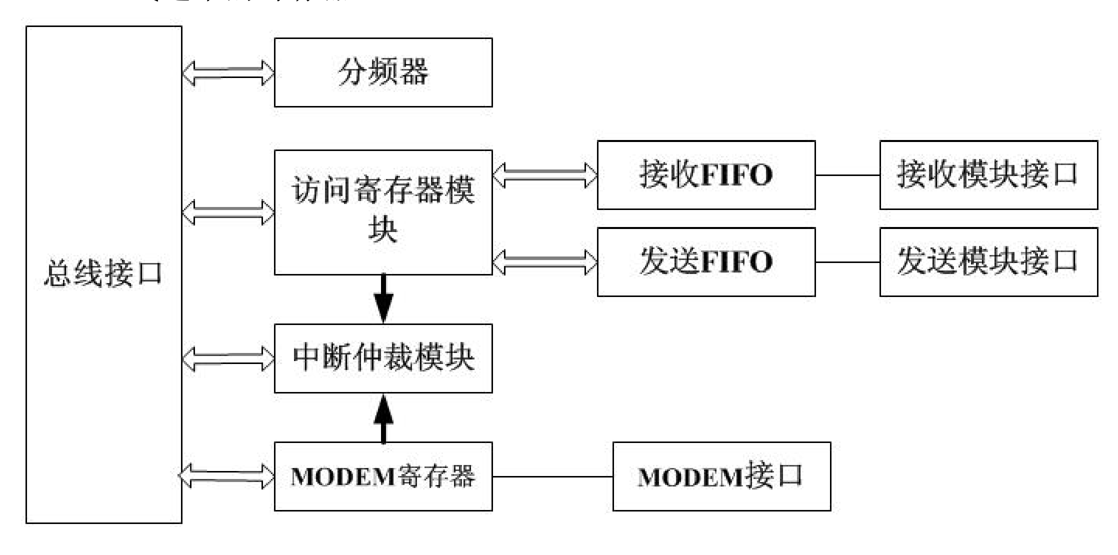


图 4-29 UART 控制器结构

UART 控制器各模块之间通过内部数据通道传输数据，有如下主要数据通

道。

- **CPU 访问寄存器：** CPU 通过 LPB 总线访问 UART，向寄存器进行读写访问控制操作。
- **CPU 向普通 UART 模块（非 MODEM）读写数据：** CPU 将要写的数据通过 LPB 总线写进 UART 模块的 FIFO 发送队列，由发送模块进行串行发送。CPU 要读的数据通过 LPB 总线从 UART 模块的 FIFO 接收队列中取出，FIFO 接收队列中的数据是由接收模块从串行接收端口接收到的，已转换为并行数据格式存入 FIFO 中。
- **CPU 向支持 MODEM 的 UART 模块读写数据：** 与普通 UART 模块不同的是，在向 UART 模块读写数据时还要考虑到 MCR 和 MSR 寄存器中的设置。CPU 发送数据前先要在 MCR 寄存器中设置相应位，通知 MODEM 计算机要发送数据，然后检测 MSR 寄存器中的 DSR 和 CTS 位，看 MODEM 是否已经准备好，准备好就可以发送数据了。当检测到数据载波时会通知 CPU 接收数据。

4.8.2 UART 控制器外部接口时序图

UART 发送数据 TXD 和接收数据 RXD 的时序如图 4-30 所示。

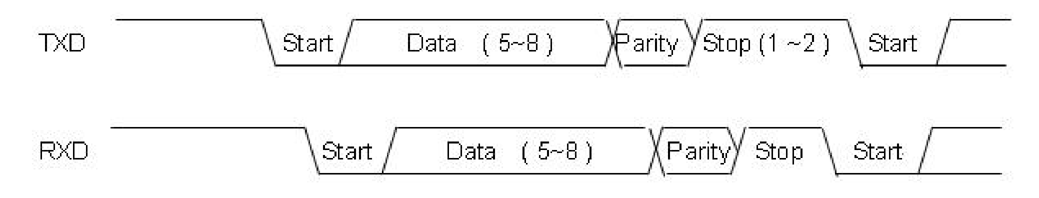


图 4-30 UART 控制器时序图

4.8.3 UART 控制器的读写操作

1. 模块初始化

- 接收和传输的FIFO被清除
- 接收和传输的移位寄存器被清除
- 打开分频器设置，给LCR寄存器写入8’b1000_0000
- 设置分频寄存器的高8位
- 设置分频寄存器的低8位
- 关闭分频器设置，恢复到正常操作，给LCR寄存器写入8’b0000_0000，LCR的低7位设置参考寄存器说明
- 重置FCR寄存器，写入8’b0000_0110，高2位设置参考寄存器说明
- 设置IER寄存器，具体参考寄存器说明

2. 模块的发送/传输操作

A. 发送数据

- 等待中断申请
- 读取 LSR，判断 LSR[6]位，为 1 表示发送数据寄存器为空
- 将数据送到 THR 中
- 对应中断位自动清零

B. 接收数据

- 等待中断申请
- 读取 IIR 寄存器，并判断对应位，参考寄存器说明
- 读接收数据寄存器 RBR
- 对应中断位自动清零

4.9 I^2C 控制器

本系统芯片集成了I^2C接口，主要用于实现两个器件之间数据的交换。I^2C 总线是由数据线SDA和时钟SCL构成的串行总线，可发送和接收数据。器件与器件之间进行双向传送，最高传送速率400kbps。

4.9.1 I^2C 控制器结构

图 4-31为 I^2C 主控制器的结构，主要模块有，时钟发生器（Clock Generator）、字节命令控制器（Byte Command Controller）、位命令控制器（Bit Command controller）、数据移位寄存器（Data Shift Register）。其余为LPB总线接口和一些寄存器。

1. **时钟发生器模块：**产生分频时钟，同步位命令的工作。
2. **字节命令控制器模块：**将一个命令解释为按字节操作的时序，即把字节操作分解为位操作。
3. **位命令控制器模块：**进行实际数据的传输，以及位命令信号产生。
4. **数据移位寄存器模块：**串行数据移位。

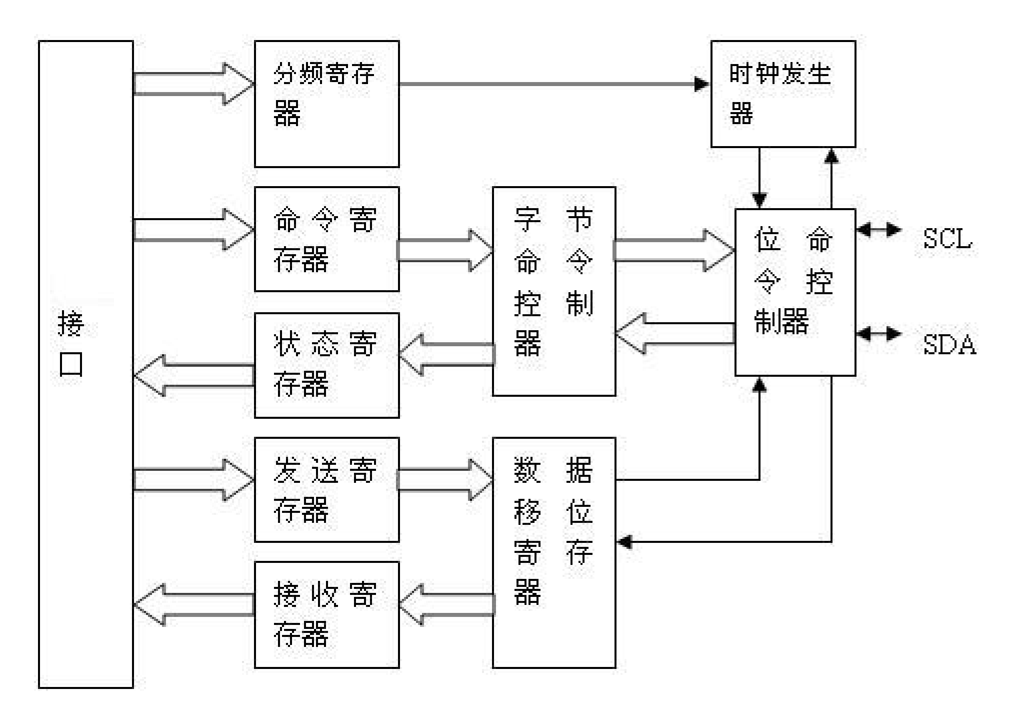


图 4-31 I²C 主控制器结构

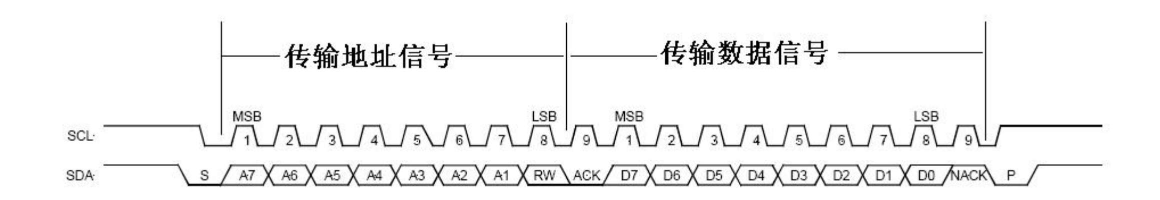
4.9.2 I²C 控制器外部接口时序图

I²C总线的标准通讯过程由4部分组成（图 4-32(a)）：

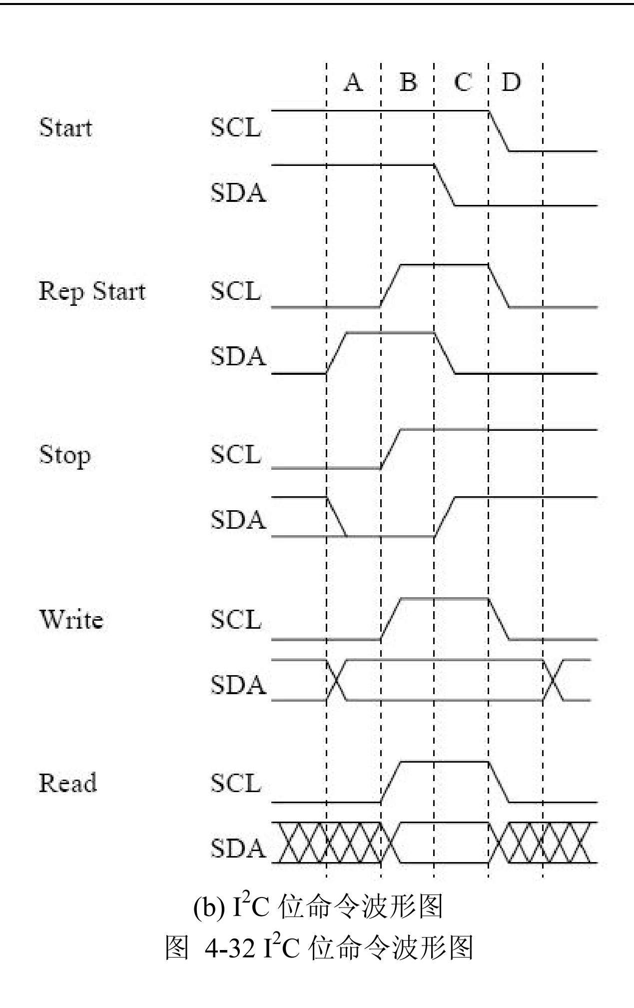
START信号产生：SCL和SDA均为高电平时，I²C总线处于空闲（idle）状态。SCL为高电平，SDA由高电平向低电平跳变，则开始新的数据传输。

1. **从地址传输**：START信号之后，主控制器传送从设备地址（7位）和RW（读/写）控制位。然后从设备返回ACK（确认）位。
2. **数据传输**：主控制器进行读/写，逐一传输字节数据（8位），每个字节后随从设备返回的ACK位。
3. **STOP信号产生**：SCL为低电平，SDA由低电平向高电平跳变，主机结束数据传输，并释放I²C总线。

图 4-32(b) 是 I²C控制器的位命令波形图，每种波形由4部分组成。其中Rep Start（Repeated START）信号为前面未产生STOP信号情况下的START信号，用于主机在不释放I²C总线下，改换从设备或传输方向。



(a) I²C 通讯过程



(b) I²C 位命令波形图

图 4-32 I²C 位命令波形图

4.9.3 I²C 控制器的读写操作

1. 模块初始化

- 设置控制寄存器CTR使控制器到配置分频寄存器模式，写入8'b0000_0000
- 设置分配寄存器的低8位
- 设置分配寄存器的高8位
- 设置控制寄存器CTR使控制器到正常模式并打开中断使能，写入8’b1100_0000

2. 模块的发送/传输操作

A. 写从设备操作

- 设置控制寄存器使控制器到正常模式，写入 8’b1100_0000
- 将从设备地址写入到发送寄存器中
- 写值到命令寄存器 CR 中，生成 start 和 write 信号，写入 8’b1001_0000
- 等待中断申请
- 读状态寄存器 SR，判断 SR[1]的值，为 1 表示传输完成
- 将从设备存储地址写入到发送寄存器中

- 写值到命令寄存器 CR 中，生成 write 信号，写入 8’b0001_0000
- 等待中断申请
- 读状态寄存器 SR，判断 SR[1]的值，为 1 表示传输完成
- 将数据写入到发送寄存器中
- 写值到命令寄存器 CR 中，生成 write 信号，写入 8’b0001_0000
- 等待中断申请
- 读状态寄存器 SR，判断 SR[1]的值，为 1 表示传输完成

B．读从设备操作

- 设置控制寄存器使控制器到正常模式，写入 8’b1000_0000
- 将从设备地址写入到发送寄存器中
- 写值到命令寄存器 CR 中，生成 start 和 write 信号，写入 8’b1001_0000
- 等待中断申请
- 读状态寄存器 SR，判断 SR[1]的值，为 1 表示传输完成
- 将从设备存储地址写入到发送寄存器中
- 写值到命令寄存器 CR 中，生成 write 信号，写入 8’b0001_0000
- 等待中断申请
- 读状态寄存器 SR，判断 SR[1]的值，为 1 表示传输完成
- 将从设备地方和读操作信号写入到发送寄存器中
- 写值到命令寄存器 CR 中，生成 start 和 write 信号，写入 8’b1001_0000
- 等待中断申请
- 读状态寄存器 SR，判断 SR[1]的值，为 1 表示发送完成
- 写值到命令寄存器 CR 中，生成 read 信号，写入 8’b0010_0000
- 等待中断申请
- 读状态寄存器 SR，判断 SR[1]的值，为 1 表示传输完成
- 读接受寄存器

3．中断处理

- 接收到中断申请
- 读状态寄存器 SR，判断 SR[1]的值，为 1 表示传输完成
- 写发送数据寄存器或者读接受数据寄存器
- 控制器自动清除中断申请

4.10 GPIO

通用输入/输出控制器GPIO控制器一般用来从芯片GPIO引脚上直接读入数

据到内部寄存其中或者将内部寄存器数据直接写到芯片引脚上。系统中有60条GPIO引脚，其中12条专用，48条全部可以与其它模块的引脚复用。因此使用前首先将相应的GPIO使能寄存器位置为1，然后从相应的GPIO_I寄存器中读取GPIO输入的值，如果要往GPIO输出，则将输出值写入对应的GPIO_O寄存器中。GPIO的使能寄存器以及输入输出寄存器定义请参见3.4.9至3.4.11，3.9.1至3.9.13章节的介绍。

4.11 Watchdog

在系统死锁时，监视器（看门狗）用于复位系统。其基本原理是必须在指定的时间内改变某个寄存器的值，否则，监视器产生复位信号，唤醒系统。具体使用时，先将看门狗的控制寄存器

WD_CTRL 的使能位置 1，复位时默认值为 0，然后设置看门狗定时寄存器

WD_TIMER的值。设置完成后，看门狗定时寄存器每拍减1，当减到0时触发复位信号，所以必须按时喂狗，即修改看门狗定时寄存器的值，保证它不等于0。相关看门狗寄存器参见3.4.13和3.4.14节。

4.12 中断控制器

SOC 内置简单、灵活的中断控制器。除了一些通过 GPIO 输入的中断信号外，中断控制器还可以管理由内部事件引起的中断。所有的中断寄存器的位域安排相同，一个中断源对应其中一位。

- 芯片各模块的中断源
  - ✓ URT1（ 1 号串口）产生的中断
  - ✓ URT0（0 号串口）产生的中断
  - ✓ Mouse（PS/2 鼠标）产生的中断
  - ✓ Keyboard（PS/2 键盘）产生的中断
  - ✓ SPI（串行外围接口）产生的中断
  - ✓ I^2C（串行通讯总线控制器）产生的中断
  - ✓ 两路 MAC（以太网控制器）产生的中断
  - ✓ LCD（LCD 控制器）产生的中断
  - ✓ 外部 4 路中断
- 中断的属性
  - ✓ 按触发类型：可以配置为电平触发或边沿触发
  - ✓ 按触发极性：可以配置为 1 触发或 0 触发

- ✓ 按 CPU 输出：可以配置为从 CPU 的 0 号中断引线输出或 1 号中断引线输出

中断的使用首先要设置中断使能寄存器中相应的位来使能该中断，系统复位时默认是不使能中断的。然后设置中断触发类型寄存器、中断极性控制寄存器和中断输出控制寄存器相应的属性。最后当发生中断时，通过中断状态寄存器查看相应的中断源。相关中断寄存器的详细定义参见3.4.2至3.4.8章节。

中断触发方式分为电平触发与边沿触发两种，电平触发方式时，中断控制器内部不寄存外部中断，此时对中断处理的响应完成后只需要清除对应设备上的中断就可以清除对CPU的相应中断。例如，上行网口向CPU发出接收包中断，网络驱动处理中断后，只要清除上行网口内部的中断寄存器中的中断状态，就可以清除CPU中断控制器的中断状态，而不需要通过对应的INT_CLR 对CPU进行清中断。但是在边沿触发的方式下，中断控制器会寄存外部中断，此时软件处理中断时，需要通过写对应的INT_CLR，清除CPU中断控制器内部的对应中断状态。另外，在边沿触发的情况下，用户可以通过写INT_SET 位强置中断控制器的对应中断状态。

4.13 JTAG 控制器

SOC3210 系列的 JTAG 控制器符合 IEEE 1149.1 标准，通过标准测试访问接口（TAP）进行边界扫描测试。SOC3210 系列中 TAP 总共包括 6 个信号：TAP_SEL 信号和 JTAG 接口标准信号 JTCK、JTMS、JTDI、JTDO 和 JRST。其中 TAP_SEL 接地时 SOC3210 系列才进入 JTAG 模式；而接口信号 JTCK、JTMS、JTDI、JTDO 和 JRST 中 4 个是输入信号接口和另外 1 个是输出信号接口。

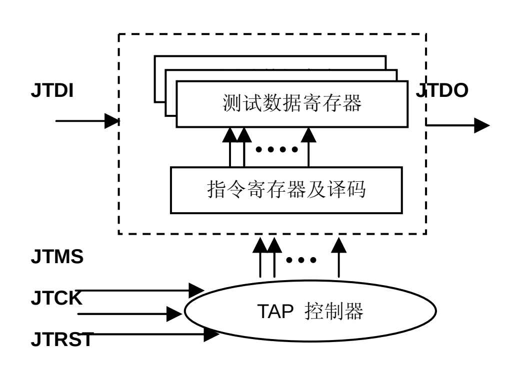


图 4-33 JTAG 测试逻辑结构

上图所示为 JTAG 控制器的标准输入输出信号，具体描述如下：

- JTCK 用于 JTAG 测试的时钟输入，为 TAP 的操作提供了一个独立的、基本的时钟信号，TAP 的所有操作都是通过这个时钟信号来驱动的。
- JRST_用于 JTAG 控制器的复位控制。
- JTMS 用于控制 TAP 状态机的转换（测试模式）。通过 JTMS 信号，可以控制 TAP 在不同的状态间相互转换。JTMS 信号在 JTCK 的上升沿有效。
- JTDI 用于串行数据的输入，所有要输入到特定寄存器的数据都是通过 JTDI 接口一位一位串行输入的（由 JTCK 驱动）。
- JTDO 用于串行数据的输出，所有要从特定的寄存器中输出的数据都是通过 JTDO 接口一位一位串行输出的（由 JTCK 驱动）。

TAP接口的具体扫描测试过程参考IEEE 1149.1标准。

5 系统特性

5.1 时钟系统

SOC3210 系列芯片有多个时钟域，详细如表 5-1 所示。包括从外部输入的晶振时钟 XIN，这是 CPU PLL 和 SYS PLL 的输入时钟。CPU_CLK 是 32 位 RISC CPU 的工作时钟，通过 CPU PLL 锁相环倍频 XIN 获得。SYS_CLK（即 SDCLK）是芯片的系统时钟，通过 SYS PLL 倍频 XIN 产生，芯片中外部存储控制器和内部高速互连总线都处于系统时钟，而低速外设控制器的时钟是通过系统时钟四分频获得，所以系统时钟是 SOC3210 系列芯片中最重要的时钟，大部分的时钟都是与其有直接或间接的关联。

表 5-1 芯片时钟域

编号	时钟	产生方式	描述	SOC 3210W	SOC 3210M	SOC 3210I
1	XIN	外部输入	Crystal 时钟，5MHz；作为 CPU PLL 和 SYS PLL 的输入，通过 CLKBAK_MODE_=1 选择（默认）	√	√	√
2	CLKBAK	外部输入	备份时钟，25MHz；作为 CPU PLL 和 SYS PLL 的输入，通过 CLKBAK_MODE_=0 选择	-	-	-
3	TCK_CLK	外部输入	JTAG 时钟	√	√	√
4	CPU_CLK	PLL 倍频	32 位 RISC CPU 时钟，150MHZ~266MHz	√	√	√
5	SYS_CLK	PLL 倍频	系统时钟，50MHz~133MHz	√	√	√
6	LCD_CLK	由 CPU_CLK 分频产生	LCD 时钟，2MHz~40MHz	-	√	√
7	BIT_CLK_I	外部输入	AC97 时钟，12.288MHz	√	√	√
8	LPB_CLK	由 SYS_CLK 四分频产生	低速外设 SPI、PS/2、UART 和 I^2C 控制器的时钟	√	√	√
9	MTX_CLK x2	由外部 PHY 的 PLL 产生	MAC 数据输出时钟，共有两个，2.5MHz/25MHz	√	√	√
10	MRX_CLK x2	由外部 PHY 的 PLL 产生	MAC 数据接收时钟，共有两个，2.5MHz/25MHz	√	√	√

32 位 RISC CPU 的锁相环的倍频是通过上电时芯片引脚 HPI_RST_、NN_CLE、EMI_OE_的上下拉进行配置，具体配置如表 5-2 所示，相应的配置

值可以通过时钟分频寄存器的 CPU_FR_CFG[5:3] （详见 3.4.1 节）读出。

表 5-2 CPU PLL 倍频的引脚上电配置

HPI_RST_	NN_CLE	EMI_OE_	CPU_FR_CFG[5:3]	XIN	CPU 时钟
下拉	下拉	下拉	0	5MHz	不用
下拉	下拉	上拉	1	5MHz	150MHz
下拉	上拉	下拉	2	5MHz	160MHz
下拉	上拉	上拉	3	5MHz	175MHz
上拉	下拉	下拉	4	5MHz	200MHz
上拉	下拉	上拉	5	5MHz	225MHz
上拉	上拉	下拉	6	5MHz	250MHz
上拉	上拉	上拉	7	5MHz	266MHz

系统时钟 SYS 的锁相环的倍频是通过上电时芯片引脚 A18、A17、A16 的上下拉进行配置，具体配置如表 5-3 所示，相应的配置值可以通过钟分频寄存器的 SD_FR_CFG[2:0]（详见 3.4.1 节）读出。当配置的 SYS_CLK 的频率为 CPU_CLK 的 1/2 时， SYS PLL 会自动关闭，并且使用 CPU_CLK 的二分频时钟作为 SYS_CLK。

表 5-3 SYS PLL 倍频的引脚上电配置

A18	A17	A16	SD_FR_CFG[2:0]	XIN	SYS 时钟
下拉	下拉	下拉	0	5MHz	不用
下拉	下拉	上拉	1	5MHz	50MHz
下拉	上拉	下拉	2	5MHz	66MHz
下拉	上拉	上拉	3	5MHz	80MHz
上拉	下拉	下拉	4	5MHz	100MHz
上拉	下拉	上拉	5	5MHz	120MHz
上拉	上拉	下拉	6	5MHz	133MHz
上拉	上拉	上拉	7	5MHz	不用

推荐使用频率：CPU 时钟 ＝266MHz，SYS 时钟 ＝100MHz。

LCD 控制器的分频时钟可以通过设置时钟分频寄存器的 [15:9]（LCD_CLK_CFG）位进行配置，实现 CPU_CLK 的 2 分频到 256 分频，即 LCD_CLK_CFG+1 分频。

低速外设控制器的分频时钟可以通过设置时钟分频寄存器的 [20:18]（LPB_CLK_CFG）位进行配置，实现同频到 16 分频，甚至时钟停止的控制，具体分频关系如表 5-4 所示。

表 5-4 分频时钟配置

时钟分频寄存器的配置值	系统时钟的分频

000	时钟停止
001	2 分频
010	4 分频
011	8 分频
100	16 分频

SOC3210 系列芯片还有六个外部输入时钟，包括一个 AC97 数据传输时钟、两个 MAC 数据发送时钟、两个 MAC 数据接收时钟和一个 JTAG 测试时钟。

5.2 系统复位

SOC3210 系列芯片在系统上电复位时，需要将某些芯片引脚上下拉进行配置。如下表格所示的全部引脚：

表 5-5 上电配置引脚汇总

配置引脚	引脚数量	功能	SOC3210W	SOC3210M	SOC3210I
HPI_RST_ NN_CLE EMI_OE_	3	确定 CPU 时钟的频率	√	√	√
A18、A17、A16	3	确定 SYS 系统时钟的频率	√	√	√
NN_ALE	1	NOR FLASH 的位宽，上拉表示 16 位，下拉表示 8 位	√	√	√
CLKBAK_MODE_	1	选择 PLL 的输入时钟，下拉选择 CLKBAK，上拉选择 XIN	–	–	–
SCAN_MODE_	1	芯片进入扫描链测试模式， 仅 CP 和 FT 时使用	–	–	–
TEST_MODE_	1	扫描链测试启动 仅 CP 和 FT 时使用	–	–	–

SOC3210 系列芯片中许多功能的引脚都复用成 GPIO，详见 2 接口定义。当芯片复位时，默认是使用各种功能而不作为 GPIO 引脚，如果需要作为 GPIO 引脚，必须先将相应的 GPIO 使能寄存器位置为 1，详见 4.10GPIO。

SOC3210 系列芯片在系统复位时，默认是屏蔽所有的中断，如果需要使用中断，必须先设置中断使能寄存器中相应的位来使能该中断，详见 4.12 中断控制器。

SOC3210 系列芯片在系统复位时，监视器（看门狗）是不工作的，如果需

要使用看门狗，必须先将控制寄存器

WD_CTRL 的使能位置 1，详见 4.11Watchdog。

SOC3210 系列芯片在系统复位时，32 位 RISC CPU 的 PC 值为 0xbfc0_0000，转换成物理地址是 0x1fc0_0000，这个地址属于 boot 的地址空间，系统复位后，32 位 RISC CPU 将从这里开始取指，所以 bios 引导程序必须放在这个位置。

另注：SOC3210 系列芯片没有 CKE 引脚，而对于 SDRAM 颗粒而言，该引脚用于使能 SDRAM 颗粒，在与本芯片的板级连接使用时，必须将颗粒的 CKE 引脚做上拉处理。

5.3 地址空间

SOC3210 系列芯片的物理地址空间一共 512MB，具体分配如表 5-6 所示。

表 5-6 SOC3210 系列芯片地址空间

起始地址	大小	描述	SOC3210W	SOC3210M	SOC3210I
0x0000_0000	256M	SDRAM	√	√	√
0x1000_0000	128M	Reserved	–	–	–
0x1800_0000	32M	NOR Flash	√	√	√
0x1e00_0000	16M	HPI	√	√	√
0x1f00_0000	32	EMI 配置寄存器	√	√	√
0x1f00_0040	4	NAND Flash 数据端口	√	√	√
0x1f00_1000	4K	LCD 配置寄存器	–	√	√
0x1f00_2000	4K	Reserved	–	–	–
0x1f00_3200	256	HSB_MISC 配置寄存器	√	√	√
0x1f00_4000	4	SPI 配置寄存器	√	√	√
0x1f00_4040	32	PS/2 配置寄存器	–	–	√
0x1f00_4080	16	UART #0 配置寄存器	√	√	√
0x1f00_4090	16	UART #1 配置寄存器	–	–	√
0x1f00_40D0	8	I^2C 配置寄存器	√	√	√
0x1f00_4100	256	LPB_MISC 配置寄存器	√	√	√
0x1f00_4200	128	AC97 配置寄存器	√	√	√
0x1f00_4280	128	AC97 DMA 配置寄存器	√	√	√
0x1f00_5200	256	MAC0 配置寄存器	√	√	√
0x1f00_5300	256	MAC1 配置寄存器	√	–	–
0x1fc0_0000	1M	Boot 空间	√	√	√
0x1fd0_0000	1M	Reserved	–	–	–
0x1fe0_0000	1M	Reserved	–	–	–

注：Boot 空间地址物理上与 NOR FLASH 最低 1M 空间相重合

5.4 特别引脚使用说明

芯片中部分信号的引脚是内置上拉电阻的，包括 JTAG 接口部分信号：

TAP_SEL、JRST_ 、JTCK、JTDI、JTMS；系统模式设置信号：SYS_RST_、【部分不引出的保留信号：SCAN_MODE_、TEST_MODE_、CLKBAK_MODE_，这三个信号可以不管】；还有 GPIO12-15 复用做中断信号 INT3~INT0。除此之外，芯片中其他全部引脚都没有内置上拉或者下拉电阻。

前面 5.2 提到的系统复位中用到跳线引脚，其所需的上拉或者下拉电阻都必须通过芯片外部的电阻来实现。如“图 5-1 需作上拉/下拉的引脚使用示”中 NN_ALE 所示的方法做 PCB 的设计，若只固定为上拉或者下拉状态，则不需要拨动开关以节省器件。

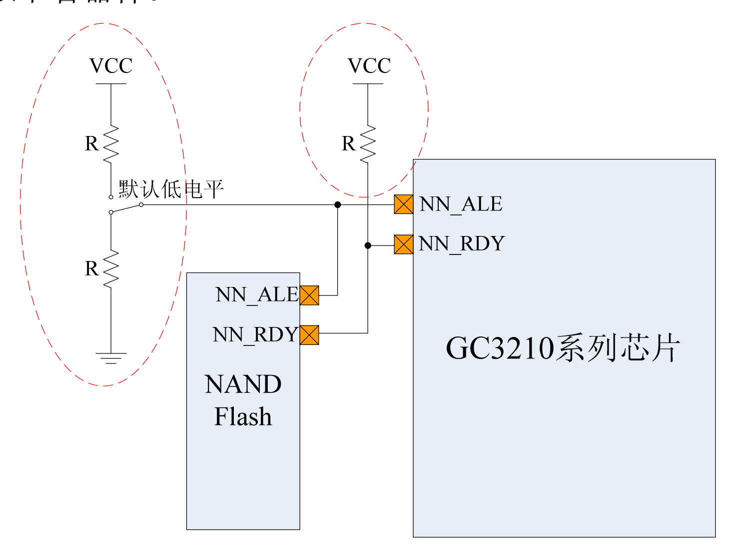


图 5-1 需作上拉/下拉的引脚使用示图

还有两个引脚需要特别注意的：

一个是 NAND Flash 的 ready 信号“NN_RDY”，它是 NAND Flash 的输出信号（即 SOC3210 系列芯片的输入信号），因为它通常是一个 OD 引脚，需要外部加上拉电阻，否则访问速度会变得很慢；

另外一个 HPI 接口的 ready 信号“HPI_RDY_”，若其所接的 DSP 如 Infineon 的 PEB 33322 的输出引脚也是 OD 类型，所以也需要外部加上拉电阻。其方法如“图 5-1 需作上拉/下拉的引脚使用示”中 NN_RDY 所示。

特别推荐：拉电阻 R 采用 10K 欧姆的阻值。

6 电气特性

6.1 临界工作参数

表 6-1 临界工作参数

参数	符号	数值			单位	说明
		最小值	典型值	最大值		
内核电压	VCCINST	-0.5		2.5	V	
IO 电压	VCCIO	-0.5		4.6	V	
PLL 电压	AVDD18 AVDD_5AP	-0.5		2.5	V	
输入电压	VI	-0.5		6	V	
输出电压	VO	-0.5		4.6	V	
存储温度	Ts	-65		150	℃	
环境温度	Ta				℃	待定
ESD 保护	Vesd	0		2000	V	HBM 模型

注意： 超出临界工作范围会对器件导致永久的损坏。长时间工作在临界范围会影响器件的可靠性，并可能导致不可逆的损坏。

6.2 推荐工作参数

6.2.1 直流电气特性

表 6-2 直流电气特性

参数	符号	数值			单位	说明
		最小值	典型值	最大值		
内核电压	VCCINST	1.62	1.8	1.98	V	
IO 电压	VCCIO	2.97	3.3	3.63	V	
PLL 电压	AVDD18 AVDD_5AP	1.62	1.8	1.98	V	独立滤波
IO 输入低电压		－0.3		1.2	V	
IO 输入高电压		1.5		5.5	V	
输入漏电流		-10			uA	
输出高电压				0.4	V	
输入高电压		2.4			V	
输入引脚电容					pF	
引脚电感					nH	
内置上拉电阻		39	65	116	KOhm	

6.2.2 交流电气特性

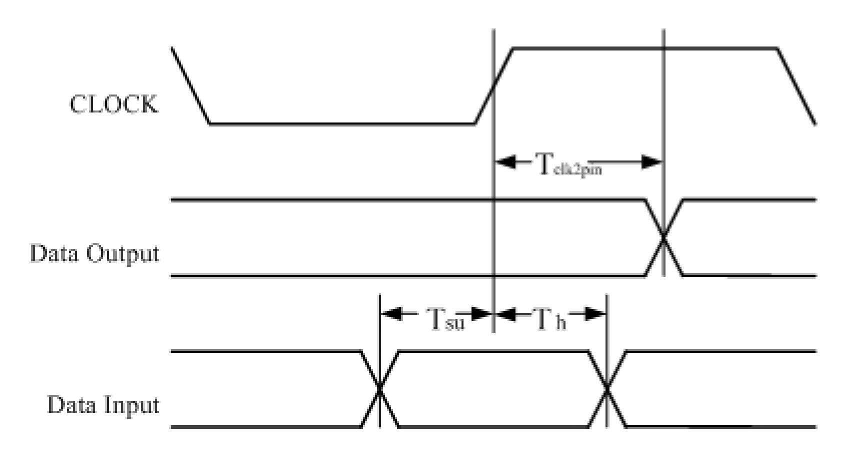


图 6-1 交流电气特性图

说明： 1) 输出延迟是从时钟输入管脚到信号输出管脚的总延迟。
2) 输出延迟的三个值分别是最好、典型、最坏三种工作条件下的最大延迟。
3) 输入保持时间的正值表示信号输入需要进行延迟补偿。

SDRAM 接口时序

时序	最小	典型	最大	单位
输出延迟（clk2pin）	3.60	5.30	8.80	ns
输入建立时间（Tsu）		2.50		ns
输入保持时间（Th）		2.50		ns

NOR FLASH 接口时序

时序	最小	典型	最大	单位
输出延迟（clk2pin）	4.05	5.90	9.60	ns
输入建立时间（Tsu）		2.55		ns
输入保持时间（Th）		1.00		ns

AC97 接口时序

时序	最小	典型	最大	单位
输出延迟（clk2pin）	4.0	6.0	10.5	ns
输入建立时间（Tsu）		4.6		ns
输入保持时间（Th）		0.35		ns

SPI 接口时序

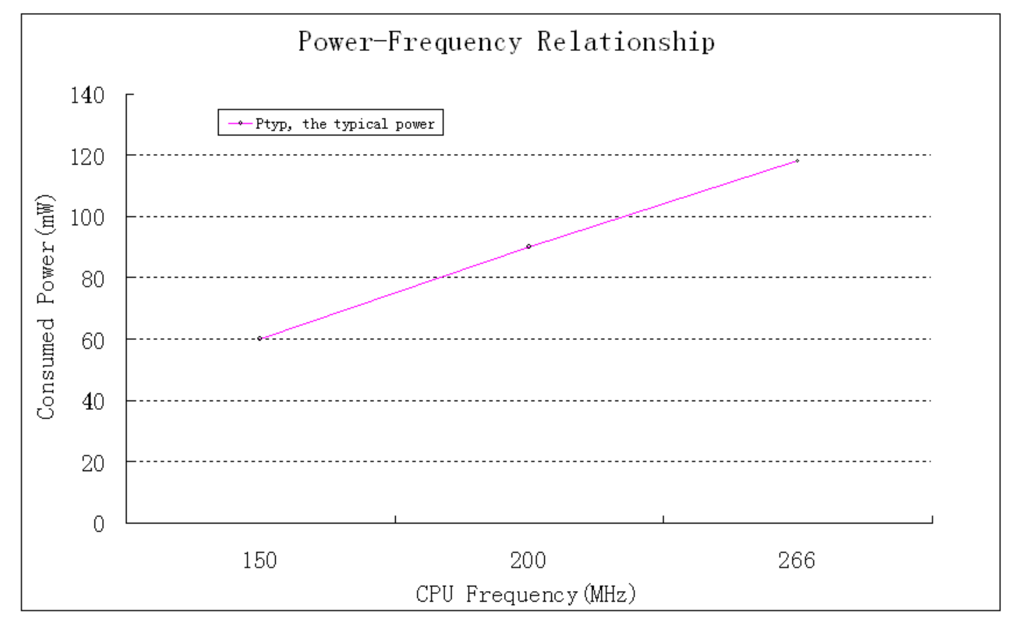
时序	最小	典型	最大	单位
输出延迟（clk2pin）	4.35	6.45	10.50	ns
输入建立时间（Tsu）		4.30		ns
输入保持时间（Th）		0.80		ns

I2C 接口时序

时序	最小	典型	最大	单位
输出延迟（clk2pin）	3.80	5.55	9.10	ns
输入建立时间（Tsu）		0.60		ns
输入保持时间（Th）		1.60		ns

6. 3 功耗特性

参数	典型值	单位	备注
动态功耗	118	mW	@266MHz
	90	mW	@200MHz
	60	mW	@150MHz



6.4 封装信息

SOC3210 系列采用多种封装形式以满足各种不同应用场合的需要，同时实现成本的最优化，封装形式包括 LQFP208 和 QFP208 封装形式。

SOC3210W	LQFP208 和 QFP208（两种封装仅厚度及引脚长度不同）
SOC3210M	LQFP208 和 QFP208（两种封装仅厚度及引脚长度不同）
SOC3210I	LQFP208 和 QFP208（两种封装仅厚度及引脚长度不同）

具体封装参数将参考各自的产品 datasheet。